Design Automation and Analysis of Three-Dimensional Integrated Circuits

by

Shamik Das

S.B. E.E., Massachusetts Institute of Technology (2000)
S.B. Mathematics, Massachusetts Institute of Technology (2000)
M.Eng. E.E.C.S., Massachusetts Institute of Technology (2000)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2004

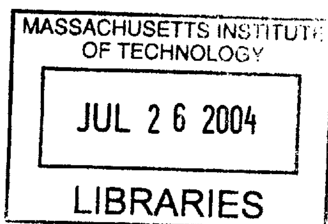
© Massachusetts Institute of Technology 2004. All rights reserved.

Author ....
Department of Electrical Engineering and Computer Science
May 1, 2004

Certified by ....
Rafael Reif
Associate Department Head and Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by ....
Anantha P. Chandrakasan
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students



BARKER

Design Automation and Analysis of Three-Dimensional Integrated Circuits

by

Shamik Das

Submitted to the Department of Electrical Engineering and Computer Science
on May 14, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This dissertation concerns the design of circuits and systems for an emerging technology known as three-dimensional integration. By stacking individual components, dice, or whole wafers using a high-density electromechanical interconnect, three-dimensional integration can achieve scalability and performance exceeding that of conventional fabrication technologies.

There are two main contributions of this thesis. The first is a computer-aided design flow for the digital components of a three-dimensional integrated circuit (3-D IC). This flow primarily consists of two software tools: PR3D, a placement and routing tool for custom 3-D ICs based on standard cells, and 3-D Magic, a tool for designing, editing, and testing physical layout characteristics of 3-D ICs. The second contribution of this thesis is a performance analysis of the digital components of 3-D ICs. We use the above tools to determine the extent to which 3-D integration can improve timing, energy, and thermal performance. In doing so, we verify the estimates of stochastic computational models for 3-D IC interconnects and find that the models predict the optimal 3-D wire length to within 20% accuracy. We expand upon this analysis by examining how 3-D technology factors affect the optimal wire length that can be obtained. Our ultimate analysis extends this work by directly considering timing and energy in 3-D ICs. In all cases we find that significant performance improvements are possible. In contrast, thermal performance is expected to worsen with the use of 3-D integration. We examine precisely how thermal behavior scales in 3-D integration and determine quantitatively how the temperature may be controlled during the circuit placement process. We also show how advanced packaging technologies may be leveraged to maintain acceptable die temperatures in 3-D ICs.

Finally, we explore two issues for the future of 3-D integration. We determine how technology scaling impacts the effect of 3-D integration on circuit performance. We also consider how to improve the performance of digital components in a mixed-signal 3-D integrated circuit. We conclude with a look towards future 3-D IC design tools.

Thesis Supervisor: Rafael Reif
Title: Associate Department Head and Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Anantha P. Chandrakasan
Title: Professor of Electrical Engineering and Computer Science

To my mother, father, and sister

and

to Anne

Acknowledgments

This dissertation would not have been possible without the love and support of my family. My mother, father, and sister have cared for me and instilled in me a sense of purpose that goes beyond mere circuits. For their dedication and inspiration, I will be forever grateful.

I am honored to have been the student of two professors, Rafael Reif and Anantha Chandrakasan, during my graduate tenure at MIT. Their mentorship of my research has been outstanding, and I could not have asked for better guides. Both advisors have given me a wealth of perspective, insight, and motivation.

Several other professors at MIT have been instrumental in my development as a scientist and engineer. I would like to thank Duane Boning for serving on my thesis committee and for the guidance he has ably provided in this capacity. John Kassakian has been an exceptional graduate counselor by keeping me on track with respect to degree requirements and giving me valuable career advice.

It is my personal (though not unique) belief that every student should participate in teaching, and I have had the fortunate opportunity to learn from a master, Professor Amar G. Bose. The experience of being one of his teaching assistants has fundamentally shaped my views on education, engineering, and communication, not to mention politics, society, religion, and the weather. For this alone, my graduate career has been worthwhile.

I would also like to thank all the members, current and former, of my two research groups. For their collaboration on this research effort and others, and for their companionship and advice, I appreciate them tremendously. My life as a graduate student, on a daily basis and at conferences, presentations, and reviews, was all the more enriched by sharing it with such colleagues. The broader community of people at the Microsystems Technology Laboratories at MIT also deserves much appreciation. A special thanks goes out to Susan Kaufman and Margaret Flaherty for their tireless administrative efforts, as well as for keeping me in tune with the world outside of work.

Several people have made specific contributions that deserve mention here. Professor Arifur Rahman of Polytechnic University developed a model that forms the basis for part of the work in this dissertation. During the few months that we were colleagues at MIT and in subsequent years, he has provided valuable guidance both in my attempts to analyze and validate his model and in my career in general. I would also like to thank MIT

students Elizabeth Basha, Katie Butler, Patrick Griffin, Wei-Han Huang, and Vivian Lei for volunteering to test one of the design tools I developed for this dissertation, as well as for agreeing to let me publish their results.

My time at MIT has not been occupied solely by the analysis of integrated circuits. Along the way, I have become blessed with many friends. My four years at Zeta Beta Tau fraternity as an undergraduate have provided me with close friendships that continue to this day. I also would not have maintained my sanity, health, and motivation were it not for the sport of ultimate and the friendships I have formed through it. I would like to thank my teammates on the MIT Ultimate Team for the experience, though I must acknowledge that my two remaining years of college eligibility provided a strong disincentive to the completion of this dissertation.

Finally, this dissertation would not exist were it not for my fiancée, Anne. She has been a constant companion throughout my graduate years, providing compassion and encouragement, as well as bringing formidable literary skills to bear on drafts of this document. I can only hope that over our life together, I can return the love she has already given me.

Contents

1 Introduction **23**

1.1 Motivation of this Work 23

1.1.1 Scaling Limitations of Conventional Integration Technology 23

1.1.2 The Potential of Three-Dimensional Integration 25

1.2 Three-Dimensional Integration Technology 26

1.2.1 Packaging Methods 27

1.2.2 Monolithic Approaches 29

1.2.3 Sample Process Flow: Copper Wafer Bonding 31

1.3 Design Tradeoffs Associated with the 3-D Integration Process Flow 33

1.3.1 Digital ICs 33

1.3.2 Analog/Mixed-Signal ICs 33

1.4 Overview of Previous Work 34

1.4.1 Stochastic Modeling of 3-D ICs 34

1.4.2 Architectural Investigation 34

1.4.3 Unresolved Problems 35

1.5 Contributions of this Dissertation 36

2 Design Tools for Three-Dimensional Integrated Circuits **39**

2.1 Overview 39

2.2 Logic Synthesis 40

2.3 Floorplanning 42

2.4 Placement 43

2.4.1 Global Placement 44

2.4.2 Detailed Placement 44

2.4.3 Placement Algorithm: Simulated Annealing 44
2.4.4 Placement Algorithm: Quadratic Placement 46
2.4.5 Placement Algorithm: Partitioning 48
2.4.6 Detailed Placement Algorithms 50
2.5 Routing 51
2.5.1 Hierarchical Approach 52
2.5.2 Global Maze Router 53
2.6 Layout 53
2.7 PR3D: The Placement and Routing Tool 54
2.7.1 3-D Standard-Cell Placement Algorithm 56
2.7.2 3-D Global Routing 58
2.7.3 Comparison of PR3D with Other Tools 59
2.8 3-D Magic: The Layout Editor 60
2.8.1 User Interface Design 60
2.8.2 Circuit Issues 61
2.8.3 Data Representation 63
2.8.4 Sample Layouts Using 3-D Magic 65
2.9 Summary 69

3 Wire-Length Performance of 3-D Integrated Circuits 71
3.1 Previous Work on 3-D IC Analysis 71
3.2 The Rahman Model 74
3.2.1 Derivation 74
3.2.2 Adaptations for Standard-Cell Circuits 76
3.3 Analysis of 3-D ICs: Model vs. PR3D 78
3.3.1 Calibration 78
3.3.2 Verification of the Rahman Model 78
3.3.3 Further Analyses via PR3D 84
3.4 Summary 88

4 Performance Characteristics of 3-D ICs 91
4.1 Overview 91
4.2 Tool Adaptations for Performance-Driven Design 93

4.3 Methodology and Circuits Under Test 95
4.4 Timing Characteristics of 3-D ICs 96
4.5 Energy Characteristics of 3-D ICs 98
4.5.1 Energy Performance of the Conventional Circuits Under Test 98
4.5.2 Energy Optimization in 3-D 100
4.6 Energy-Delay Product 104
4.7 Summary 104

5 3-D IC Thermal Management and Optimization 107
5.1 Motivation 107
5.2 First-Order Model for Die Temperature in 3-D ICs 109
5.3 Placement-Based Optimization of Thermal Characteristics 111
5.4 Thermal Characteristics of 3-D ICs 112
5.5 Active Cooling Using Microchannels 118
5.5.1 First-Order Model 120
5.5.2 Modifications to the Thermal Algorithms 122
5.5.3 Placement-Based Analysis 123
5.6 Summary 127

6 Future Considerations for 3-D Integration 131
6.1 Overview 131
6.2 Predictive Technology Models: Impact of 3-D Integration in Future Technology Generations 131
6.2.1 Motivation 131
6.2.2 Fixed-Chip Scaling 132
6.2.3 3-D Integration of the Projected "Largest Chip" 137
6.3 Opportunities for Mixed-Signal 3-D Integration 139
6.3.1 Overview 139
6.3.2 Optimization for Digital Performance in Mixed-Signal Systems . . . 141
6.3.3 Optimization of the Digital Noise Impact on Analog/RF Subsystems 143
6.4 Architecture for a Design Flow for Mixed-Signal 3-D ICs 146
6.5 Summary 149

7 Conclusion **151**

7.1 Summary of Research Results 151

7.2 Directions for Future Work 153

7.2.1 Technology Research 153

7.2.2 CAD Tools 154

7.2.3 Circuit Design 155

A Usage Information for the 3-D Design Tools **157**

A.1 PR3D: The Placement and Routing Tool 157

A.1.1 Platform Support 157

A.1.2 Usage 157

A.1.3 File Formats 159

A.2 3-D Magic: The Layout Editor 161

A.2.1 Platform Support 161

A.2.2 Usage 161

A.2.3 Commands 162

A.2.4 Extensions to the Magic Technology File Format 163

List of Figures

1-1 Projected inverter FO4 and 1-mm interconnect delays for various technology nodes. . . . 24

1-2 Schematic of a 3-D integrated circuit with interleaved device layers and inter-layer interconnects. . . . 25

1-3 Wire-length distribution of a typical circuit as a function of number of device layers used. . . . 26

1-4 (a) Vertical multi-chip module (MCM-V) schematic. (b) Schematic of flip-chip bonded circuit. . . . 27

1-5 Vertical multi-chip module (MCM-V) showing inter-layer interconnect back-plane. Left: schematic; right: package photo (reprinted from [1]). . . . 28

1-6 Flip-chip package with solder-bump interconnect (reprinted from [2]). . . . 29

1-7 Wafer-bonded structure with two device layers and copper interconnect interface. (Figure courtesy A. Fan, MIT.) . . . 30

1-8 Multiple-wafer structure using oxide as the bonding interface. The inter-wafer interconnects are formed after bonding. (Figure courtesy MIT Lincoln Laboratory.) . . . 31

1-9 Handle-wafer attachment, grindback, via formation, and copper patterning steps of the wafer bonding process. (Figure courtesy A. Fan.) . . . 31

1-10 Thermocompression and handle release steps of the wafer bonding process. (Figure courtesy A. Fan.) . . . 32

2-1 Simplified flowchart for the automated design of 2-D and 3-D digital integrated circuits. . . . 40

2-2 Wire length as a function of fan-out for a benchmark circuit. . . . 41

2-3 Wire length as a function of fan-out (low fan-out cases only). . . . 42

2-4 Typical simulated-annealing sequence for a simple network at initial, intermediate, and final stages. 45

2-5 Single-net example of the hierarchical routing procedure. Routing proceeds from stage (a) to (f) by recursive partitioning. 52

2-6 Partitioning strategy where plane assignment is done first in order to minimize the number of inter-plane vias. 56

2-7 Partitioning strategy where plane assignment is done by considering aspect ratio in order to minimize total wire length. 56

2-8 For small inter-wafer via sizes, we permit same-row interconnects to be split among multiple wafers. For large inter-wafer via sizes, we partition into wafers before reaching the single-row block size. 57

2-9 Screen shot of 3-D Magic exhibiting a two-wafer circuit layout. 62

2-10 Bonded stack of `CellDef` structures with `up` and `down` pointers for front-side and back-side bonding contacts and `prev` and `next` pointers for stack traversal. 64

2-11 Bottom wafer of a two-wafer class-E amplifier designed by Wei-Han Huang and Vivian Lei. 65

2-12 Top wafer of a two-wafer class-E amplifier designed by Wei-Han Huang and Vivian Lei. 66

2-13 Power efficiency of the 1.9 GHz amplifier in 2-D ($\circ$) and 3-D ($*$) implementations. Total power is given in third curve (Δ). 67

2-14 Crosstalk on adjacent multiplexer lines in the selector subcircuit of the 1.9 GHz amplifier, in 2-D (Δ) and 3-D ($*$) cases, as a function of separation distance. 67

2-15 Block diagram for a four-bit ADC designed by Elizabeth Basha, Katie Butler, and Patrick Griffin. 68

2-16 Top wafer (left) and bottom wafer (right) of the two-wafer ADC designed by Elizabeth Basha, Katie Butler, and Patrick Griffin. 68

2-17 Signal-to-noise-and-distortion ratio (SNDR) for 2-D and 3-D implementations of the ADC. 69

3-1 N-leaf planar fat-tree network exhibiting $O(\sqrt{N})$ bisection bandwidth. . . . 72

3-2 Schematic representation of the derivation of occupancy distribution: $N_a = 1$ is the logic gate in question, N_c is the number of target logic gates at Manhattan distance l gate pitches, and N_b is the number of logic gates in between. t_x, t_y, and t_z are the gate width, height, and inter-layer thickness, respectively, in micrometers. (Figure courtesy A. Rahman.) 74

3-3 Predicted wire-length distribution for the ibm14 benchmark circuit with inter-layer pitch t_z of 1 micrometer. 79

3-4 Placed wire-length distribution for the ibm14 benchmark circuit with inter-layer pitch t_z of 1 micrometer. 80

3-5 Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is given relative to the 2-D placed wire length. Inter-layer pitch t_z is 1 micrometer. 81

3-6 Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is normalized to exhibit the percentage reduction due to 3-D integration. Inter-layer pitch t_z is 1 micrometer. 81

3-7 Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is given relative to the 2-D placed wire length. Inter-layer pitch t_z is 250 micrometers. 82

3-8 Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is normalized to exhibit the percentage reduction due to 3-D integration. Inter-layer pitch t_z is 250 micrometers. 82

3-9 Predicted percentage of interconnects that span multiple device layers, compared with placement and routing data for $t_z = 1$ and $t_z = 250$. 84

3-10 Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from placement. Total wire length is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire. 85

3-11 Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from routing. Total wire length is minimized by the routing tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire. 86

3-12 Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from placement. The number of inter-layer vias is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire. . . . 86
3-13 Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from routing. The number of inter-layer vias is minimized by the routing tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire. 88
3-14 Length of the longest wire (as a function of number of device layers) for various inter-layer via capacitances. Total wire length is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire. 89
3-15 Length of the longest wire (as a function of number of device layers) for various inter-layer via capacitances. The number of inter-layer vias is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire. 89
3-16 Total wire length (as a function of number of device layers) of the ibm03 benchmark circuit, using vias vs. using flip-chip solder bumps for the inter-layer interconnect. 90

4-1 Power consumption for a high-performance microprocessor at various technology generations. 92
4-2 Delay model for gates and wires. 95
4-3 Cycle time of an FFT datapath using various placement modes. 96
4-4 Cycle time of a DES implementation using various placement modes. 97
4-5 Cycle time of a 64-bit MAC using various placement modes. 97
4-6 Energy consumption of an FFT datapath in timing-optimized vs. timing-constrained placement. 98
4-7 Energy consumption of a DES chip in timing-optimized vs. timing-constrained placement. 99
4-8 Energy consumption of a multiplier-accumulator chip in timing-optimized vs. timing-constrained placement. 100

4-9 Energy consumption of the FFT datapath vs. number of wafers used for placement. 100
4-10 Energy consumption of the DES chip vs. number of wafers used for placement. 101
4-11 Energy consumption of the 64-bit MAC vs. number of wafers used for placement. 101
4-12 Energy-delay product for the FFT datapath vs. number of wafers used for placement. 102
4-13 Energy-delay product for the DES chip vs. number of wafers used for placement. 103
4-14 Energy-delay product for the 64-bit MAC vs. number of wafers used for placement. 103
4-15 Wire energy-delay product for the FFT datapath vs. number of wafers used for placement. 104
4-16 Wire energy-delay product for the DES chip vs. number of wafers used for placement. 105
4-17 Wire energy-delay product for the 64-bit MAC chip vs. number of wafers used for placement. 105

5-1 Minimum required heat sink thermal resistance by technology generation, based on ITRS projections for microprocessor size and power dissipation. The desired maximum die temperature is 100°C. 108
5-2 Temperature of the uppermost die in a 3-D stack, assuming 50 W power dissipation, 2 sq. cm. total circuit area, and 25°C ambient temperature. . . 111
5-3 Celsius die temperature of the top wafer of a three-wafer placement of the FFT datapath. 113
5-4 Energy distribution of the top wafer of a three-wafer placement of the FFT datapath. 113
5-5 Die temperature of the FFT datapath vs. number of wafers (fixed-die case). 114
5-6 Absolute temperature differential of the FFT datapath vs. number of wafers (fixed-die case). 114
5-7 Average-temperature z-axis differential of the FFT datapath vs. number of wafers (fixed-die case). 115

5-8 Die temperature of the FFT datapath vs. number of wafers (scaled-die case). 115

5-9 Absolute temperature differential of the FFT datapath vs. number of wafers (scaled-die case). 116

5-10 Average-temperature z-axis differential of the FFT datapath vs. number of wafers (scaled-die case). 116

5-11 Interconnect energy dissipation of the FFT datapath vs. number of wafers in energy-optimized and gradient-optimized cases. 117

5-12 Minimum required heat sink thermal resistance by technology generation, based on ITRS projections for microprocessor size and power dissipation and 3-D performance-scaling data from this work. The desired maximum uppermost-die temperature is 100°C. 118

5-13 Wafer-bonded structure with the addition of fluid microchannels for cooling (c.f. Figure 1-7). 120

5-14 Microchannel with fluid flow in the positive x direction, power flow profile $P(x)$, and fluid temperature $T_{ch}(x)$, in an ambient solid temperature T_{die}. . 121

5-15 Celsius die temperature prediction for the 2-D FFT, with microchannel heat sink, as a function of channel cross-sectional dimension and fluid velocity. . 124

5-16 Head loss in p.s.i. for the FFT microchannels as a function of channel cross-sectional dimension and fluid velocity. 125

5-17 Die temperature of the FFT datapath vs. number of wafers (microchannel case). 126

5-18 Absolute temperature differential of the FFT datapath vs. number of wafers (microchannel case). 126

5-19 Average-temperature z-axis differential of the FFT datapath vs. number of wafers (microchannel case). 127

5-20 Celsius die temperature as a function of the number of wafers and the number of microchannels used. The 2-D version of this chip dissipates 50 W and has dimensions 1.5 cm × 1.5 cm. The microchannels are 50 μm in effective diameter and the water flow is 25 cm/s at 25°C at the inlet. 128

6-1 Cycle time of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling. 133

6-2 Energy consumption of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling. 134
6-3 Energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling. 134
6-4 Interconnect energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling. 135
6-5 Cycle time of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling. 135
6-6 Energy consumption of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling. 136
6-7 Energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling. 136
6-8 Interconnect energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling. 137
6-9 Predicted CPU frequency for several technology generations using one to five device layers for implementation. 138
6-10 Substrate noise spectrum for a 1 GHz Pentium® 4 microprocessor operating at 1.5 V supply and dissipating 15 Watts (reprinted from [3]). 140
6-11 Placement of a 3-D mixed-signal system. In (a) each module is targeted for a separate wafer. In (b) non-critical digital components are placed on memory or analog wafers in order to reduce wasted silicon. 140
6-12 Three implementations of a mixed-signal circuit. Top left: single wafer; top right: two wafers with digital circuitry isolated to bottom wafer; bottom: two wafers with equal footprint. 142
6-13 Cycle time of the FFT datapath in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias. 142
6-14 Interconnect energy dissipation of the FFT datapath in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias. 143

6-15 Cycle time of the 64-bit MAC in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias. 144
6-16 Interconnect energy dissipation of the 64-bit MAC in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias. 144
6-17 Cycle time of the 64-bit MAC two-wafer, equal-area, mixed-signal implementation. (1) and (3) are cases where the clock is distributed over both wafers; (2) and (4) are cases where the clock is restricted to the bottom wafer. (1) and (2) are cases where the inter-wafer vias are small; (3) and (4) represent larger inter-wafer vias. 145
6-18 Interconnect energy dissipation of the 64-bit MAC two-wafer, equal-area, mixed-signal implementation. (1) and (3) are cases where the clock is distributed over both wafers; (2) and (4) are cases where the clock is restricted to the bottom wafer. (1) and (2) are cases where the inter-wafer vias are small; (3) and (4) represent larger inter-wafer vias. 146
6-19 Digital vs. proposed mixed-signal design flow paradigms. 146
6-20 Outline of a candidate mixed-signal design flow. 147

List of Tables

1.1 ITRS predictions for circuit performance. 24

1.2 ITRS predictions for wires in integrated circuits. 24

2.1 Algorithm 3DPLACE for multi-wafer placement using min-cut partitioning. 55

2.2 Effective number of bits (ENOB) for 2-D and 3-D implementations of the ADC. 67

3.1 Performance of our placer and other state-of-the-art placers on the IBM-PLACE 2.0 circuit benchmark set. Wire lengths are in meters. 78

3.2 Cells of the ISPD '98 benchmark suite used in this study. 79

3.3 Absolute prediction error relative to placed wire length as a function of number of device layers and inter-layer thickness. 83

3.4 Absolute prediction error relative to routed wire length as a function of number of device layers and inter-layer thickness. 83

3.5 Placement and routing data for the ISPD '98 benchmark suite. Wire lengths are in μm. Percentages are reductions relative to the one-wafer case. 87

4.1 Relevant parameters for the circuits in this study. 96

6.1 Properties of devices and mid-level interconnect in 180 nm and 35 nm technologies [4]. 132

Chapter 1

Introduction

1.1 Motivation of this Work

1.1.1 Scaling Limitations of Conventional Integration Technology

For several decades, integrated circuits have profoundly impacted our everyday lives. In order to sustain this impact, it is widely expected that the decades-long trend of exponential growth in circuit performance and functionality must be sustained as well. However, the path to continued growth contains many obstacles.

The International Technology Roadmap for Semiconductors (ITRS) provides a detailed plan for achieving this growth [5]. In Table 1.1, we see specifically what is desired of circuit designers and manufacturers. The performance demands listed in the table must be met both by increasing transistor device capabilities and by improving the performance of the wires that connect these devices.

While device scaling is by no means a solved problem, the performance of scaled devices is at least understood to increase as desired. In contrast, the performance of scaled wires does not increase similarly. Table 1.2 shows the degree to which interconnect must be shrunk merely to meet functionality demands. However, at this level of scaling, worst-case and even average-case interconnect performance *decreases* with each generation.

Figure 1-1 illustrates the problem. A fan-out-of-four (FO4) inverter (i.e. an inverter that is used to drive four identical inverters) scales with increasing technology generations, such that the signal delay through an FO4 inverter is roughly proportional to the node length. However, the delay through a representative 1 mm wire increases exponentially from gen-

technology node (nm)	180	130	90	65	45	35
microprocessor transistors/chip (millions)	21	76	226	453	773	1,227
on-chip local clock frequency (GHz)	1.25	2.1	4.171	9.285	15.079	20.065
chip-to-board clock frequency (GHz)	1.2	1.6	2.5	4.883	9.536	14.901
power supply (V)	1.8	1.5	1.2	1.1	1.0	0.9
CPU power (W)	90	130	158	189	218	240
chip size (mm^2)						
at introduction	280	280	280	280	280	280
in production	140	140	140	140	140	140

Table 1.1: ITRS predictions for circuit performance.

technology node (nm)	180	130	90	65	45	35
number of metal layers	6-7	7-9	10-14	11-15	12-16	12-16
minimum metal pitch (nm)	360	300	214	152	108	84
effective resistivity ($\mu\Omega \cdot$ cm)	2.2	2.2	2.2	2.2	2.2	2.2
effective inter-layer dielectric constant	3.5-4.0	3.3-3.6	3.1-3.6	2.7-3.0	2.3-2.6	2.3-2.6

Table 1.2: ITRS predictions for wires in integrated circuits.

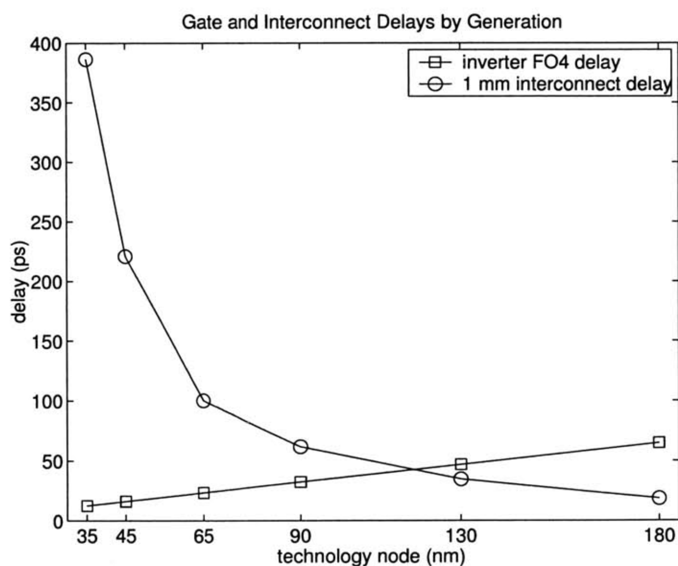


Figure 1-1: Projected inverter FO4 and 1-mm interconnect delays for various technology nodes.

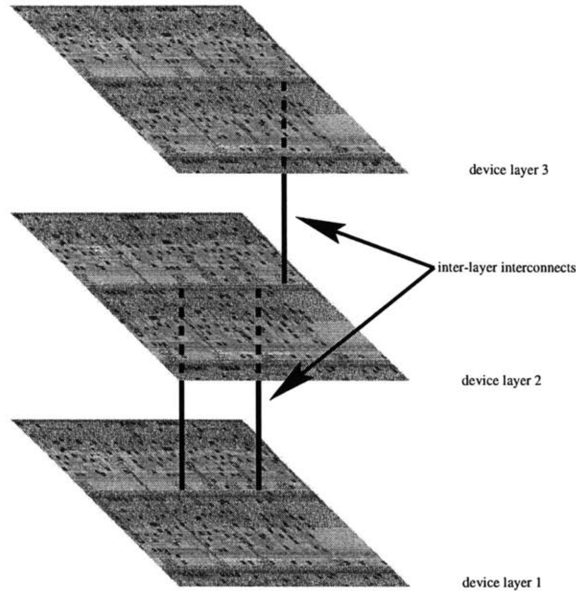


Figure 1-2: Schematic of a 3-D integrated circuit with interleaved device layers and inter-layer interconnects.

eration to generation, due to the increased resistance from scaling down the cross-sectional area of the wire. More importantly, since we expect the size of maximum-functionality circuits to hold steady or even increase, we cannot even improve performance by scaling down the length of our representative 1 mm wire as we increase the technology generation.

1.1.2 The Potential of Three-Dimensional Integration

Three-dimensional integration aims to alleviate the above scalability issues. A **three-dimensional integrated circuit** (3-D IC) is any circuit in which the active devices are not confined to a single plane. We may consider such a circuit to be a collection of distinct 2-D (conventional) ICs, each of which individually is called a "device layer" [6], "tier" [7], "stratum" [8], or simply a "wafer" (although the latter term does not strictly apply in some technologies). These conventional layers, together with a means of interconnecting devices on separate layers, make up a three-dimensional integrated circuit. A schematic rendition of such a circuit is given in Figure 1-2.

At first glance, it is clear that 3-D integration offers greater device density for a given footprint area. What is not clear is how 3-D integration may affect other circuit metrics such as speed and energy consumption. The first indication of what may be achieved in a technologically-feasible 3-D IC lies in the work of A. Rahman *et al.* [8,9]. This work analyzes

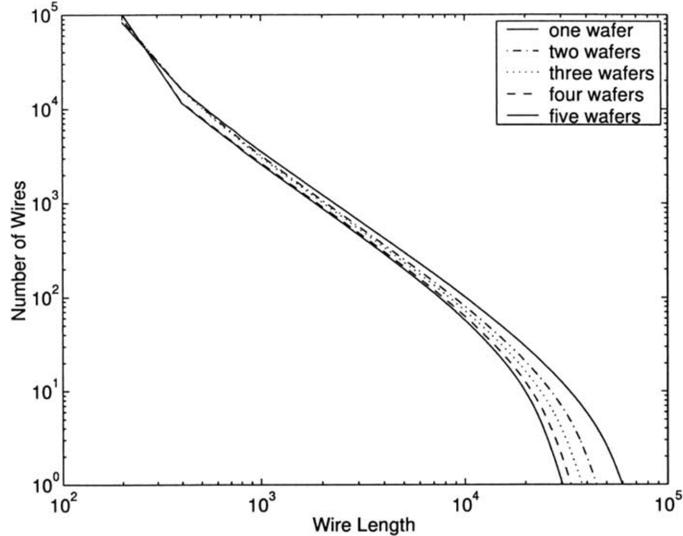


Figure 1-3: Wire-length distribution of a typical circuit as a function of number of device layers used.

the distribution of wires in a general circuit according to their length; it finds that in a wide class of 3-D integration technologies, the wire-length distribution shifts in response to an increase in the number of device layers as shown in Figure 1-3. The leftward shift in wire-length distribution is the mechanism by which three-dimensional integration aims to improve circuit performance since the longer wires in any such distribution disproportionately affect cycle time, energy consumption, and routability.

While the general behavior exhibited in Figure 1-3 may be characteristic of 3-D integration, the precise scale and separation of the distributions are what result in specific performance improvements. These particular aspects are highly dependent on the choice of technology itself. For this reason, we must first seek to understand what characterizes a potential three-dimensional integration technology.

1.2 Three-Dimensional Integration Technology

There are many technologies that can be described, however loosely, as three-dimensional. The fundamental traits underlying these technologies are that active devices may be stacked in multiple layers and that the scalability of circuit dimensions along all three axes is not inherently limited. These various technologies may be classified as either packaging technologies, by which three-dimensionality is achieved after the individual 2-D chip components

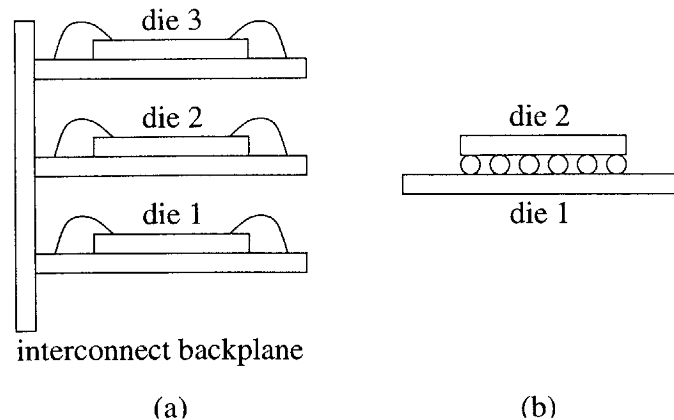


Figure 1-4: (a) Vertical multi-chip module (MCM-V) schematic. (b) Schematic of flip-chip bonded circuit.

have been fabricated, or monolithic technologies, by which the full 3-D structure is formed prior to packaging. In all cases, understanding how the 3-D technology parameters will affect circuit performance is the ultimate goal.

1.2.1 Packaging Methods

The first packaging technology capable of forming three-dimensional circuits is the vertical multi-chip module (MCM-V) [10, 11]. In an MCM-V package, individual dice are fabricated and bonded to printed-circuit-board (PCB) backplanes. The input and output pads are wire-bonded to connections on the surface of the PCB. The separate PCBs are then connected to a high-bandwidth interconnect backplane that serves as the communication infrastructure between the dice. Figure 1-4(a) gives a schematic view of the structure of a generic MCM-V package, and Figure 1-5 shows a candidate MCM-V technology [1, 12, 13]. The principal trade-off associated with this type of package is that while its manufacturing does not involve any unusually complicated processing steps, the resulting inter-layer interconnect is neither high-performance nor low-latency compared with wires on the individual chips.

Two approaches that attempt to overcome this performance limitation to some degree are ultra-thin chip stacking [14, 15] and multilayer thin-film packaging (MCM-D) [16]. In these technologies, individual dice are prepared, stacked, and bonded using a benzocyclobutene (BCB) polymer spin-on. The preparation stage involves whole-wafer thinning (down to 10-15 μm) before die cut; stacking is performed with an alignment accuracy of

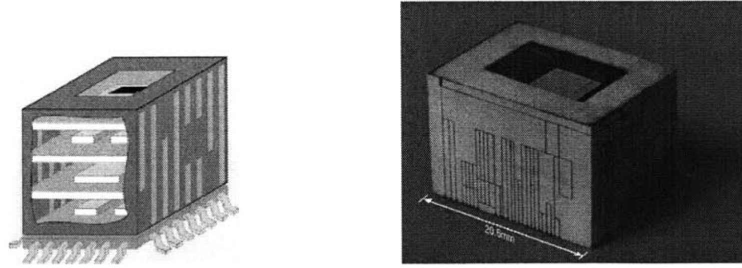


Figure 1-5: Vertical multi-chip module (MCM-V) showing inter-layer interconnect backplane. Left: schematic; right: package photo (reprinted from [1]).

±10 μm. Once the dice are bonded, they are wired to a surrounding ring of routing tracks for inter-layer interconnection.

These technologies offer better performance than MCM-V due to their somewhat lower-latency inter-layer interconnect. At the same time, they simultaneously offer a degree of design simplicity since the inter-layer interconnect is at the periphery. However, as with MCM-V, the inter-layer communication occurs through the periphery. This interconnect thus exhibits lower performance compared to within-die wires.

An alternative approach that potentially can be used to create 3-D ICs with higher-performance inter-layer interconnect is known as flip-chip bonding or chip-scale packaging [17]. Typically used for direct mounting of circuit substrates onto PCBs, the flip-chip method is nonetheless capable of being used as a 3-D integration technology. In flip-chip bonding, the upper surface of a die is patterned with a solder-bump interconnect. The mating surface on the PCB is patterned with pads. The die is then "flipped" onto the PCB and bonded using the solder bumps. Figure 1-4(b) shows a schematic, and Figure 1-6 shows a sample solder-bump array.

Since no fundamental constraint exists requiring the use of a PCB as the mating surface, the flip-chip approach can be used to bond two dice together. Furthermore, a platform has been suggested by which several small, customized, high-performance dice are flip-chip bonded to a larger moderate-performance die in order to integrate various high performance technologies without significant fabrication cost or compromised performance. Of course, this technique is not immediately scalable to stacks more than two dice thick; some form of through-die interconnect must be developed for such cases. Furthermore, while the solder-bump interconnect performance exceeds that of bond wires (and thus the MCM-V interconnect backplane), it still lags behind the performance of on-chip interconnect.

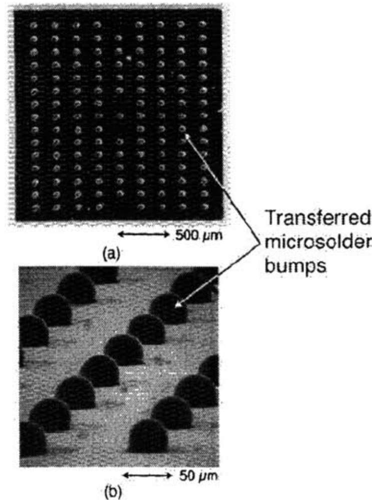


Figure 1-6: Flip-chip package with solder-bump interconnect (reprinted from [2]).

1.2.2 Monolithic Approaches

The goal of monolithic 3-D integration is to overcome the scalability and performance limitations of the aforementioned packaging methods. Thus, all such integration approaches attempt to use wafer-level fabrication techniques to build device and interconnect layers directly on top of the existing conventional plane of transistors.

The first two such techniques are epitaxy and solid-phase recrystallization. In an epitaxial 3-D integration process, silicon seed openings are fabricated alongside transistors in a conventional single-plane process. These seeds are then used to grow transistors on top of the existing devices and metallization [18]. While significant density improvements have been shown by fabricating actual circuits using this technique, it is not clear how to scale the process to more than two active layers. In a solid-phase recrystallization process, amorphous silicon is deposited on an existing integrated circuit; this silicon is then recrystallized using a laser. The resulting silicon islands may be used to produce polysilicon thin-film transistors. Thus, while this technique is highly scalable, it does not yield high-performance devices on the upper device layers, and its use is restricted to high-density memories [19, 20].

The remainder of the monolithic approaches may be classified under the term "wafer bonding." The individual wafers in such a 3-D IC are fabricated using conventional means and fused together with an inter-wafer electrical and mechanical interconnect. Wafer-bonding methods differ in terms of the bonding material and the order of fabrication oper-

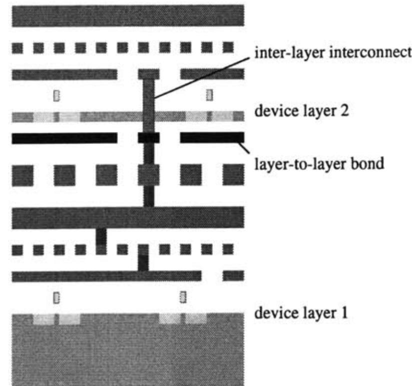


Figure 1-7: Wafer-bonded structure with two device layers and copper interconnect interface. (Figure courtesy A. Fan, MIT.)

ations. The bonding interface may be either metal or dielectric; the individual wafers may be fabricated in parallel or sequentially.

The MIT method, for example, is a copper-bonded parallel approach [21]. Front-end and back-end processing are done separately on the individual wafers that make up a given 3-D IC. The bottom-most wafer is typically a bulk silicon wafer, 500-700 μm thick, in order to provide structural rigidity; subsequent wafers are silicon-on-insulator (SOI), 1-2 μm thick, to provide scalability and high-performance interconnect. A diagram of a copper-bonded two-wafer structure is shown in Figure 1-7.

In contrast, the MIT Lincoln Laboratory method uses oxide bonding in its parallel approach [7]. The individual wafers are processed (front end and almost all back end) before bonding. Formation of inter-wafer interconnects is the remaining back-end step. This occurs after bonding since the use of oxide as a bonding material prevents the formation of ohmic contacts as a result of the bond (although capacitive, i.e. AC-coupled, inter-wafer communication has been proposed, as in [22]). Inter-wafer interconnects are formed as vias that are etched through the the entire metallization stack of the top wafer. Thus, a greater routing-area penalty is incurred; additionally, there are more stringent alignment requirements due to the nature of the via formation. Figure 1-8 shows a multiple-wafer structure using this bonding methodology.

Researchers at Rensselaer Polytechnic Institute have developed a similar process [23]. In this method, a dielectric polymer glue, e.g. BCB, is used in place of oxide bonding. The remaining process steps are essentially the same.

The Cornell University process, on the other hand, is sequential [24]. Specifically, after a

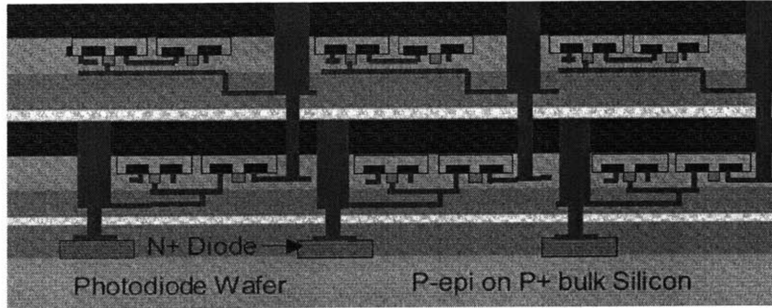


Figure 1-8: Multiple-wafer structure using oxide as the bonding interface. The inter-wafer interconnects are formed after bonding. (Figure courtesy MIT Lincoln Laboratory.)

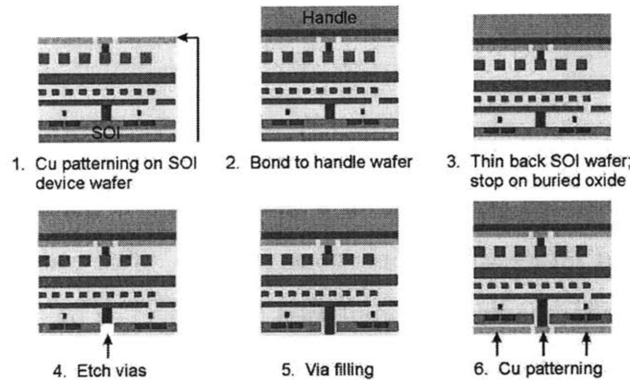


Figure 1-9: Handle-wafer attachment, grindback, via formation, and copper patterning steps of the wafer bonding process. (Figure courtesy A. Fan.)

given wafer has been processed, a blank wafer is bonded to it. The inter-wafer interconnects are fabricated together with first-level metal. Since the bonding wafer is blank, there are no alignment concerns during bonding, which results in potentially smaller inter-wafer interconnects when compared with any of the previous process technologies. However, the trade-off is that the finished bottom wafer must now endure the processing steps required to fabricate devices and wires on the blank wafer that has already been bonded to it.

1.2.3 Sample Process Flow: Copper Wafer Bonding

In order to understand the design trade-offs that arise from 3-D integration technology, it is useful to examine a sample process flow. We outline here the copper wafer bonding process of A. Fan *et al.* [21].

Figure 1-9 shows the pre-bonding process steps. This process starts with an existing wafer or stack of already-bonded wafers. To this stack we wish to bond another wafer, for which we use a typical SOI substrate (100 nm silicon with 400 nm buried oxide). This sub-

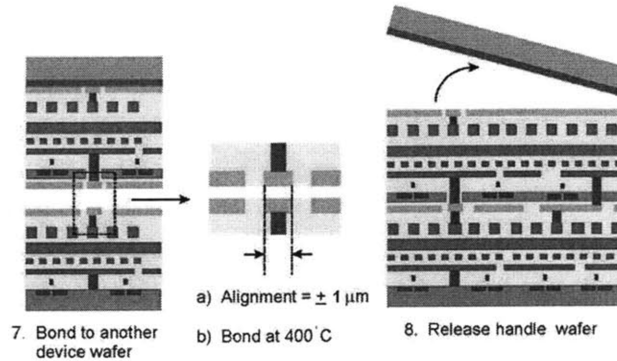


Figure 1-10: Thermocompression and handle release steps of the wafer bonding process. (Figure courtesy A. Fan.)

strate is essentially a finished circuit, as it contains all the desired devices and interconnect. If this wafer is to be bonded again (i.e. to a third wafer), it is first metallized to produce its half of the required inter-wafer connections (step 1). The wafer is then attached to a handle that is used for mechanical manipulation (step 2). The bulk silicon is then removed from the wafer (step 3); this involves a combination of mechanical grindback and chemical etching. Inter-wafer connections to the existing stack are then formed (steps 4-6). Via formation in these steps is a conventional process technique; thus, the resulting vias can be as narrow as 0.25-0.5 μm with an aspect ratio of 2:1.

In Figure 1-10, we show the bonding and handle-wafer release steps. The bonding process itself (steps 7a and 7b) is done at 350°C and 4000 mbar for 30 minutes. After bonding, the stack is annealed in nitrogen ambient for an additional 30-60 minutes. Wafer alignment is the critical process step. Both wafer-to-wafer alignment and bonding are performed in an Electronic Vision EV 450 Aligner and AB1-PV Bonder. The system has an inherent ±3 μm alignment tolerance, resulting in a copper-bonding pad pitch of at least 6 μm. Thus, wafer-to-wafer alignment is the ultimate factor in determining the inter-layer via density. With better optical alignment systems, it is possible to decrease the copper pad size down to approximately 0.5 to 1 μm, which corresponds to a substantial increase in via density. For the remainder of this dissertation, we will assume that this via density can be achieved.

The process flow iteration is completed by releasing the handle wafer (step 8). The resulting stack is ready for either packaging or subsequent bonding of additional wafers.

1.3 Design Tradeoffs Associated with the 3-D Integration Process Flow

Having illustrated the process flow, let us now consider the circuit-design trade-offs that arise. The copper-wafer bonding process previously outlined introduces distinct opportunities and challenges for both digital and mixed-signal 3-D integration.

1.3.1 Digital ICs

In a multi-layer digital system, the system components must be partitioned among the various layers. Thus, performance of the inter-layer interconnect is the process characteristic of primary interest.

In some of the packaging approaches described above, such as MCM-V or ultra-thin chip stacking, inter-layer wires must be routed to the periphery of individual layers before the wires may cross from layer to layer. As a result, the bandwidth and density of these wires are limited.

In contrast, some packaging technologies and all monolithic approaches offer a higher-density interconnect that may be fabricated at the local level. The trade-off between these technologies lies in the specific parasitic values associated with the interconnect; these may range over several orders of magnitude from copper-bonded approaches to solder-bump-interconnect technologies.

In addition, the choice of integration technology may affect signal-coupling issues. The adjacency of two substrates to a given set of metal layers reduces the amount of charge sharing between adjacent metal lines [25]. The extent to which this coupling is reduced depends on the effective capacitance between the given metal lines and the second substrate (introduced by 3-D integration). Higher substrate capacitance reduces inter-symbol interference at the expense of increasing overall capacitive energy dissipation.

1.3.2 Analog/Mixed-Signal ICs

Three-dimensional integration also provides benefits and challenges for mixed-signal and mixed-technology circuits. In analog circuits, the inter-wafer interface may be used to isolate functional units [25]. Depending on the choice of technology, or even the use of metal vs. dielectric in a specific wafer-bonding technology, the degree of isolation may be

affected significantly.

3-D integration also allows for the incorporation of multiple fabrication technologies within a single circuit or package. For example, silicon CMOS may be integrated with SiGe or InP analog, or logic-optimized CMOS may be integrated with CMOS optimized for SRAM, DRAM, or high-voltage non-volatile memories. This type of integration presents unique opportunities for circuit design; however, the integration of a small number of relatively large, discrete macro blocks in a single circuit also presents some unique design partitioning and optimization issues.

1.4 Overview of Previous Work

1.4.1 Stochastic Modeling of 3-D ICs

The bulk of prior work on 3-D integrated circuits has been in system-level stochastic modeling. Numerical models have been derived that estimate the wire-length distribution in circuits implemented in various forms of 3-D integration technology [8,26–29]. The bulk of this form of analysis has resulted in plots of the form shown in Figure 1-3.

Extensions to these models have considered specific 3-D IC technology optimizations such as variable inter-wafer distance [30]. Other ventures in the area of numerical modeling concern specific performance issues such as heat generation [31,32]. The remaining work along these lines has been in numerical modeling of specific circuit architectures in 3-D.

1.4.2 Architectural Investigation

In addition to numerical analysis of general-purpose circuits targeted for 3-D integration, several specific circuit architectures have been ported to candidate 3-D IC technologies. The prime candidates for 3-D integration explored thus far have been imagers and sensors, microprocessors, and field-programmable gate arrays (FPGAs).

Imager circuits consist of a two-dimensional array of optical sensors together with circuitry to process and deliver the sensed images off-chip. In circuits such as [33], benefit from 3-D integration is due to the fact that in conventional implementations, there is a per-pixel overhead for the processing and delivery circuitry. In a 3-D implementation, the additional wafers can be dedicated for the non-sensing components. As a result, a greater pixel density can be achieved.

A similar density impact is to be gained in FPGAs [9,34]. Like imagers, FPGAs consist of a regular array of elements. In this case, the elements are programmable functional units (typically a logic function with four to six inputs and one or two outputs, together with optional registers or tri-state drivers) and the overhead consists of wires and programmable switchboxes used to interconnect the functional units. However, with FPGAs the interconnect may consume as much as 90% of the total circuit area. The benefit of 3-D integration is that the extra routing resources in the third dimension can be used to reduce the number of conventional routing tracks required, thus increasing the density of functional units as well as shortening the wires used to connect them.

In microprocessors, a number of architectural improvements have been proposed to exploit 3-D integration [35,36]. In general, the microprocessor has been analyzed as a logic-memory system; performance enhancement is achieved either by (1) partitioning both logic and memory subsystems to reduce the logic latency as well as the memory latency, or (2) increasing the memory capacity of the system. In [35] it was determined that microprocessor instructions-per-cycle (IPC) could be increased by 20% to 30% using two-wafer integration. Furthermore, at current technology nodes, long-wire delay in microprocessors could be reduced by a factor of 2.5 to 5. Finally, it was predicted that in future technology nodes, opportunities for increased memory subsystem performance due to 3-D integration would significantly increase performance as measured by IPC.

1.4.3 Unresolved Problems

The above avenues of prior research still leave open a number of problems. First, in the area of stochastic modeling, is the question of validity: without any analysis of placed and routed circuits, it is impossible to verify that the models' predictions are correct. In fact, the models themselves vary greatly in terms of their analyses of 3-D IC performance – due in part to varied technology assumptions and to intrinsic issues of model accuracy. Of more direct importance along this line of investigation is actual circuit performance. Without having vetted predictive models for 3-D circuit wire length, it is very difficult to make reasonable predictions for circuit timing and energy consumption in three dimensions.

Second, in the area of architectural investigation, the opposite problem arises. In this area, specific opportunities for 3-D integration have been identified. However, it is not known to what extent the improvements in these circuits can be leveraged in general.

Third, in either of the above cases, it is desirable to make further circuit-based analyses of 3-D ICs. Issues such as thermal performance and technology scaling have yet to be addressed completely.

It is clear that what is needed is the ability to analyze actual circuits in a variety of 3-D implementations. Furthermore, this analysis must be carried out in a general-purpose manner, independent of architecture.

1.5 Contributions of this Dissertation

This dissertation makes two overall contributions to the understanding of 3-D integration. The first is a computer-aided design flow for 3-D ICs; the second is the performance analysis of digital 3-D ICs and IC components.

We present our design flow and algorithmic details of the tools in this flow in **Chapter 2**.

Our analysis of circuit performance begins with an adaptation of the stochastic models mentioned in Section 1.4.1 for a set of benchmark circuits used throughout the dissertation. We analyze the wire-length performance of these circuits through the use of the models and compare this data with measurements from placements generated by our tools (**Chapter 3**). We proceed to expand upon the predictions of the models by utilizing specific placement-based analyses. Having established the wire-length behavior of 3-D ICs, we develop a placement-based characterization of circuit timing and energy performance (**Chapter 4**).

We bring our design tools to bear on a significant problem in 3-D ICs: heat generation and removal (**Chapter 5**). With the use of placement-based analyses, we verify prior numerical simulations of thermal effects in 3-D ICs. Furthermore, we characterize thermal behavior in two placement contexts by demonstrating how placement-based thermal optimization can be utilized to obtain more acceptable behavior in exchange for reduced performance in other metrics. We also consider in detail the use of advanced heat-removal technologies and develop design models and guides for the implementation of such technologies within a 3-D system.

We also examine some speculative issues regarding the future of 3-D IC design (**Chapter 6**). First, we consider how the performance improvements due to 3-D integration might scale in conjunction with conventional technology scaling. Second, we explore how to expand the design flow to include mixed-signal integration. In the context of mixed-signal

integration, we examine how digital performance may be improved, and we also evaluate some methods for reducing the noise impact of these digital circuits in mixed-signal 3-D ICs. Finally, we propose a design-flow architecture for mixed-signal 3-D integrated circuits.

Chapter 2

Design Tools for Three-Dimensional Integrated Circuits

2.1 Overview

The design of a digital integrated circuit typically proceeds from a high-level specification of what the circuit is supposed to do by successively refining this specification down to the function of each individual transistor. Refining from specification to transistor layout may be done all at once; however, for all but the smallest circuits, this is intractable for both humans and computers. Thus, the design process is divided into steps such as those shown in the left half of Figure 2-1. Our goal is to identify which components of this design flow must be replaced or altered to design three-dimensional integrated circuits.

As seen in Figure 2-1, several steps are taken to produce fabrication data from a high-level specification. We take this specification to mean a behavioral or functional description in a hardware description language such as VHDL or Verilog. Thus, the first step is typically **logic synthesis**, whereby a gate-level circuit net list is determined. A **floorplan** is developed, and given the net list and physical parameters of the individual logic gates, the circuit gates are **placed** in an optimal location on the die. The resulting placement is wired or **routed**. The placed-and-routed circuit **layout** is analyzed to ensure that if fabricated according to the design, it will function according to the specification. These

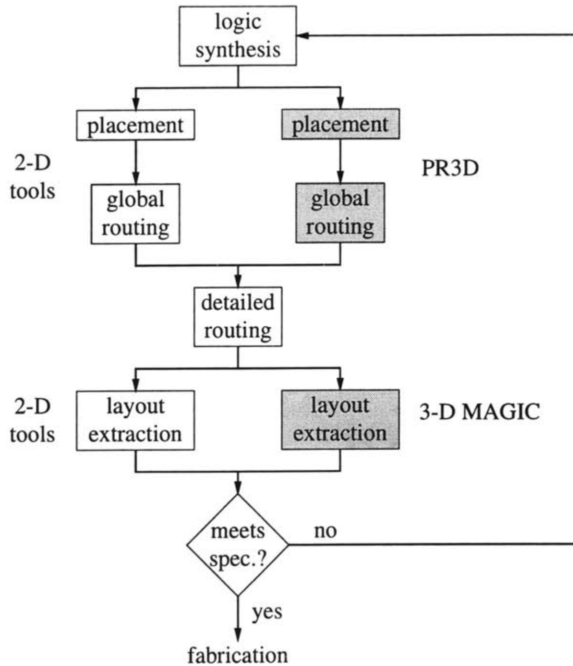


Figure 2-1: Simplified flowchart for the automated design of 2-D and 3-D digital integrated circuits.

three components – synthesis, placement, and routing – constitute the front end of physical design of digital circuits.[1]

As indicated in the right half of Figure 2-1, at several stages of the flow it is required or desired to modify the tools to design for three-dimensional integration. In the next several sections, we will address when conventional tools may be used, what changes may be required for such tools, and what tools we have developed to enable 3-D IC design.

2.2 Logic Synthesis

Logic synthesis remains for the most part a *technology-independent* phase of the design flow. The output of logic synthesis is a gate-level description of a circuit; the functionality provided by the gates themselves is independent of how these gates are fabricated. Thus, it is not strictly necessary to modify this stage of the design flow to create 3-D ICs.

However, some optimizations exist that take advantage of technology-dependent information. For example, gate vendors may offer various speed and power options for individual gates [37]. Additionally, these gates perform differently under varying input and output con-

[1]For the purposes of proper scoping, the *back end* of design, including components such as reliability, yield, and other such post-layout analyses, will not be addressed in this thesis.

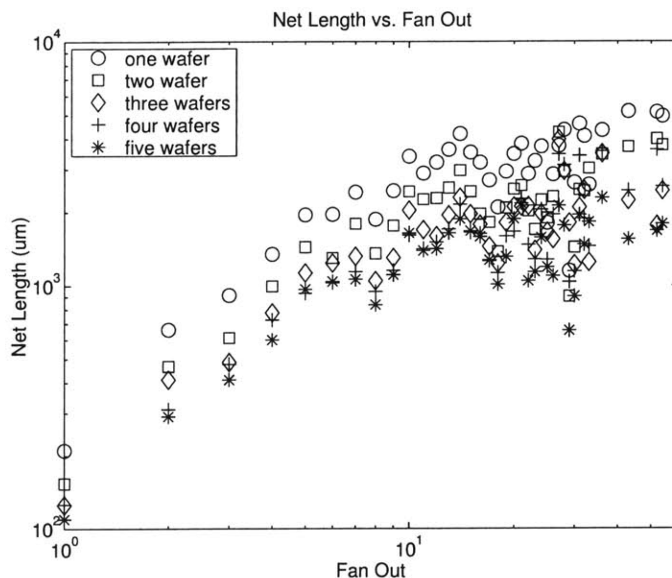


Figure 2-2: Wire length as a function of fan-out for a benchmark circuit.

ditions; an optimizing logic synthesizer may choose gates that have sufficient drive strength so as to meet design constraints. The effect of interconnect on the performance of the cells is typically captured through the use of wire-load models.

In the context of logic synthesis, wire-load models predict the capacitance of a given wire based on the number of terminals [38]. The synthesis tool uses this information to size and/or duplicate logic cells to meet specified timing or energy constraints. In 3-D ICs, we expect that the wire-length distribution will be shifted; therefore, we may capture this information in a wire-load model. Figure 2-2 shows how the wire-length-vs.-fan-out behavior changes for a benchmark circuit as we increase the number of wafers. In Figure 2-3 we see specific behavior for low-fan-out cases; it is typical to restrict logic synthesis to the generation of low-fan-out nets only. In both figures we see that there may be a use for customized wire-load models for 3-D ICs.

For two reasons, however, we choose not to implement wire-load modeling for 3-D integration. The first is that the effectiveness of wire-load modeling in deep-submicrometer designs is hotly debated [38]. More fundamentally, our ultimate goal is to explore the impact of 3-D integration on circuit performance metrics such as cycle time. If we choose to incorporate 3-D awareness at the logic-synthesis stage, we are in effect trading off primary performance improvements for improvements in circuit topology or secondary circuit metrics.

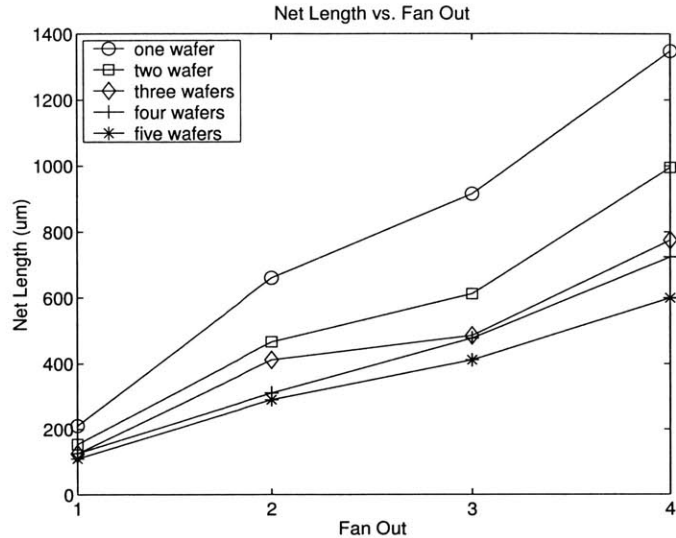


Figure 2-3: Wire length as a function of fan-out (low fan-out cases only).

For example, consider a circuit with a cycle-time constraint of 3 ns. In a conventional design flow, we would synthesize logic for a single-wafer implementation using this constraint. We could then place and route this logic using two or more wafers to obtain further improvement in cycle time or energy consumption (or both). If instead we utilize a wire-load model for, say, three-wafer integrated circuits, we would then obtain a synthesized design that meets the 3-ns constraint using three wafers. Relative to the synthesized logic for the single-wafer implementation, this logic would either occupy less area or require less intra-cell energy dissipation, depending on the optimization priority schedule given to the synthesis tool. However, we cannot subsequently improve the cycle time using multi-wafer placement and routing.

For this reason, as well as in consideration of the fact that placement algorithms have different levels of effectiveness on different topologies, we will utilize the same synthesized logic for single-wafer and multi-wafer implementations.

2.3 Floorplanning

In the design of large circuits, the hierarchical nature of the synthesis methodology results in a top-level architecture comprising a small number of large functional blocks. Circuits incorporating memories, for example, are usually partitioned into logic and memory subsystems rather than distributing the memory throughout the chip. As a result, it is sometimes

necessary to devise a **floorplan** for the chip in which locations for these few large blocks are determined prior to placement and routing of the logic subsystems.

Prior work on 3-D IC design has included automated floorplanning [39]. While the circuits considered (part of the MCNC benchmark suite [40]) are small by modern standards, this floorplanner was able to exhibit significant performance improvement in terms of total length of global wires and the length of the longest wire. However, a full determination of circuit performance requires the placement optimization of flat (i.e. non-hierarchical) circuit topologies. Flat placement optimization requires the use of different CAD tools; thus, it is on these tools that we will focus our efforts.

2.4 Placement

To simplify some of the computational aspects of the placement process, many custom circuit designers adopt the standard-cell paradigm. In this paradigm, the individual logic gates, registers, and other components are synthesized as cells of fixed height and variable width. Since the cells are of fixed height, the placement area may be defined as a number of fixed-height rows, and the placement process therefore becomes the discrete (integer) problem of assigning a row and site (location within the row) to each of the cells.

Historically, the placement process would be followed by a row-spacing determination; specifically, empty space between the rows would be allocated for routing wires, and the quantity of this space would be determined once the associated routing problem was well-defined. The spacing requirement that results could yield a sub-optimal placement, such that multiple iterations would be needed to obtain the best performance. This *variable-die* placement context has given way to a more common *fixed-die* context in modern deep-submicrometer design. In the modern context, since a large number of metal layers is available, the row spacings are fixed *a priori* (often to zero), and the routing is done over the cells.

The growing size of standard-cell circuits has motivated the development of hierarchical (top-down) placement tools. In top-down placement, the design first undergoes a **global placement** stage, during which the locations of individual cells are refined to a modest number of partitions of the entire die area. Each partition is small enough that it can be placed in a tractable manner. **Detailed placement** is then used to determine the final

locations of cells within each partition.

During both the global and detailed placement stages, it is possible to introduce awareness of three dimensions to the algorithms.

2.4.1 Global Placement

The global placement stage is devoted to refining the placement of cells to some localized area. A final location for any cells at this stage is not desired. Global placement is thus reserved for cases in which the number of cells makes direct solution intractable.

Several algorithms, described below, are suitable for global placement, since discrete locations will not be determined. In considering a global algorithm for 3-D integration, however, the relatively small number of device layers provides direction for the choice of algorithm. As will be discussed in the following sections, we will need to choose an algorithm that allows us to localize cells to any given wafer, even during the earliest stages of global placement.

2.4.2 Detailed Placement

Once global placement is complete, cells in the individual circuit partitions must then be fixed to specific locations. This is the task of detailed placement. The algorithms described in the following sections are suitable to varying degrees. In the case of 3-D IC placement, it is necessary for the algorithm to be able to localize cells to specific device layers.

2.4.3 Placement Algorithm: Simulated Annealing

Simulated annealing [41] is a method of global and detailed placement that is based on the physical process of annealing. As an algorithm for objective-function minimization, it is an extension of a generalized Monte Carlo method for simulating the states of an n-body system [42].

In this scheme, the state variables $S = \{s_i | i = 1 \dots n\}$ are the positions of the n cells in the circuit, and the objective $E(S)$ is typically the total wire length of the circuit, but may be some other metric to be minimized. This objective is analogous to the energy of the n-body system. A free variable called the temperature, T, is used to dictate how the state evolves. Specifically, the system is started in an initial configuration $S(T = T_0)$ at a high temperature T_0. A number of randomized state changes are then attempted. Each of

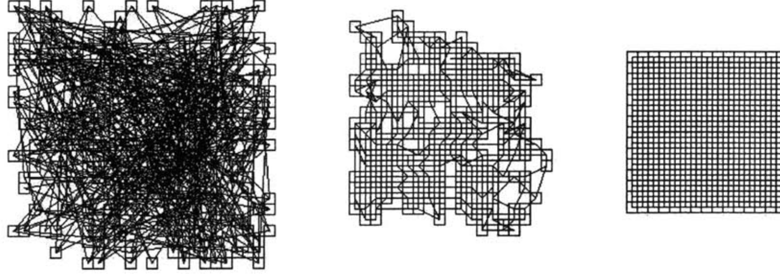


Figure 2-4: Typical simulated-annealing sequence for a simple network at initial, intermediate, and final stages.

these changes is accepted in turn if the change of state reduces the energy E. If the energy is increased due to a state change, the change is still accepted with probability $e^{-\Delta E/kT}$, where ΔE is the increase in energy and k is a constant. At the end of this sequence of state changes, the temperature is reduced and the process is repeated.

For placement in particular, the choice *annealing schedule*, or sequence of temperatures $T = T_{0,1,2,\ldots}$, strongly affects the quality of the final placement. Furthermore, no general algorithmic way of choosing a good schedule is known. Development of a useful placement tool based on simulated annealing thus rests on the determination of an acceptable schedule. Figure 2-4 shows three temperature slices in a typical simulated-annealing sequence. Fast convergence to a neighborhood of the optimal solution is exhibited here as a characteristic of a useful schedule. In contrast, schedules that do not approach a good solution before the temperature falls too low typically exhibit "lattice cracks" or "quenching," similar to the physical annealing process.

As for the state change, it typically consists of the movement a cell to a new location or the swapping of a pair of cells. Since it is intractable to consider all possible moves or swaps, practical implementations restrict choices to those moves that have a high likelihood of acceptance [41].

In considering simulated annealing as a placement algorithm, it is important to note that the algorithm is more effective for smaller placement sizes. Thus, simulated annealing is usually considered as a detailed placement tool or, in hierarchical placement strategies, as a means of incrementally improving placement quality between steps of the hierarchy.

For 3-D placement, one strength of simulated annealing is its adaptibility to many kinds of objective functions. An existing 2-D placement algorithm using simulated annealing thus may easily be adapted to three dimensions. Furthermore, the run time for a multiple-

wafer placement of a given circuit is not expected to be longer than that for a single-wafer placement. Conversely, the only direct control that can be exerted on the placement is through the modification of the energy function E. As a result, it is difficult to examine different 3-D placement strategies, such as minimum-via-count vs. minimum overall wire length, using simulated annealing.[2] In addition, the desire to implement a 3-D placer for use in large circuits imposes a requirement for a more scalable algorithm for global placement.

2.4.4 Placement Algorithm: Quadratic Placement

Quadratic placement methods are characterized by the minimization of the *squared* wire length of the placement. While this is not usually the desired metric for optimization, this choice of metric is made because there exist well-understood methods for obtaining a provably-optimal (though invalid) solution. The placement algorithm thus combines a quadratic solver with a legalization method.

The placement problem is formulated as follows: for n cells at locations (x_i, y_i) with $i = 1, \ldots, n$, the total squared wire length may be written as

$$L = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right], \tag{2.1}$$

where c_{ij} is the weight associated with the connection between nodes i and j, if this connection exists, and zero otherwise. We may reformulate this as

$$L = x^T B x + y^T B y, \tag{2.2}$$

where x and y are n-element cell-position vectors and $B = D - C$, where $C = [c_{ij}]$ and D is a diagonal matrix with d_{ii} related to the weighted degree of node i. Subject to an appropriate constraint, this equation may be solved for x and y, thus yielding a placement.

Several constraint methodologies exist. In the absence of fixed terminals, we may use a Lagrangian formulation [44]: we set

$$x^T x = y^T y = 1, \tag{2.3}$$

to produce a placement over the square $[0, 1] \times [0, 1]$. As a result, L is minimized when x

[2]However, multiple-objective formulations do exist [43].

and y are eigenvectors of B. We choose the second and third smallest eigenvalues and their corresponding eigenvectors for the placement.[3]

This formulation easily extends to three dimensions by taking the next smallest eigenvalue and corresponding eigenvector for the z axis. However, not only does this require additional computation, but the z-axis solution is also difficult to legalize, as we will address shortly.

In the more relevant case in which fixed terminals are present, a useful formulation is implemented in the GORDIAN placement tool [45]. Here, the matrix C is defined as before, except that we restrict the formulation to movable cells. Considering for now only the x dimension (since the problem is separable), we add a vector term d and a scalar f to account for the diagonal D and the connections to fixed terminals:

$$L_x = x^T C x + d^T x + f. \tag{2.4}$$

We then apply a constraint

$$Ax = u, \tag{2.5}$$

which specifies that the n cells are to be placed over q partitions of the placement area, and that in each partition, the center of gravity of the cells in that partition should be the geometric center of the partition. This constraint formulation reduces the dimensionality of the problem from n to $n - q$ since the location of one cell in each of the q regions is fixed by the locations of the remaining cells. The resulting objective may be written as

$$L'_x = x_f^T Z^T C Z x_f + c^T x_f, \tag{2.6}$$

where Z represents the dimensional reduction using A, c is the reduction of d, and x_f is the position vector of the $n - q$ free cells.

Since $Z^T C Z$ is symmetric and positive definite, this objective is minimized when

$$\frac{1}{2} Z^T C Z x_f + c = 0. \tag{2.7}$$

The locations of the movable cells may then be determined using an iterative technique

[3]The smallest eigenvalue is zero and corresponds to the solution where all cells are placed at (0.5, 0.5), since the problem is underconstrained.

such as the conjugate gradient method.

A popular modification of this technique is **force-directed** placement. The name arises from the solution to the unconstrained wire-length minimization:

$$\frac{1}{2}Cx + d = 0. \tag{2.8}$$

If we imagine that the nets connecting cells are springs, such that the force pulling cells together is proportional to the distance separating them, then Equation 2.8 represents the spring equilibrium, where the net x component of the spring forces on each cell is zero. Under this interpretation, it is straightforward to introduce additional forces of the form $e^T x$ to the system [46]. Such forces may be used to incorporate additional system constraints, such as the requirement that cells not overlap.

Two drawbacks to these formulations are that since they are quadratic programming problems, the solution is (1) in a continuous space, whereas the desired solution is in a discrete space, and (2) minimal for the quadratic objective, whereas a minimum linear wire length is desired. The former is especially problematic for 3-D placement, since in the third dimension, a highly-discrete placement is required. In particular, these algorithms tend to produce a high degree of cell overlap in the center of the placement area, which must then be resolved by iterative-improvement techniques. For this reason, quadratic placements are often used as initial solutions for a partitioning algorithm, which we describe in the next section.

2.4.5 Placement Algorithm: Partitioning

One methodology used for placement is recursive min-cut partitioning [47]. In min-cut partitioning, a circuit or sub-circuit is divided into two parts of roughly equal area such that the number of wires crossing from one part to the other is minimized.

Formally, the circuit or sub-circuit is represented as a hypergraph $H = (V, E)$, where V is a set of vertices representing the standard cells and $E \subseteq 2^V$ is a set of hyperedges with a one-to-one mapping of hyperedges to nets in the circuit. Each vertex $v \in V$ is assigned a weight $w(v)$ equal to the width of the cell, and each hyperedge $e \in E$ may be assigned a weight $w(e)$ (though this is typically taken to be $w(e) = 1$). A two-way *partitioning* is then

defined to be a map

$$p : V \mapsto \{0, 1\}. \tag{2.9}$$

The *partitions* themselves are called 0 and 1; $p(v)$ for some v may be fixed to 0 or 1 if v is an I/O pin or immovable cell (or in the case of sub-circuits, if the cell is external to the sub-circuit [48]). The partitioning is called *valid* if it satisfies a *balance criterion* on the sums-of-weights

$$W_i = \sum_{p(v)=i} w(v) \tag{2.10}$$

such as $|W_0 - W_1| \leq \tau(W_0 + W_1)$, where τ is called the *tolerance.*

We define

$$c(e) = \begin{cases} w(e) & \exists v_1, v_2 \in e | p(v_1) = 0 \wedge p(v_2) = 1 \\ 0 & otherwise \end{cases} \tag{2.11}$$

as the *cut weight* of edge e. In other words, $c(e)$ is the weight $w(e)$ if e contains vertices in both partitions, and zero otherwise. The *cut* of partitioning p is defined as

$$c(p) = \sum_{e \in E} c(e). \tag{2.12}$$

A min-cut partitioning of H is thus a valid partitioning p with the least cut $c(p)$.

The problem of determining a two-way min-cut partitioning is NP-complete [49]; there are several heuristic algorithms. The vast majority of these are based on the Fiduccia-Mattheyses (FM) algorithm [50], which is itself an efficient variation of the Kernighan-Lin algorithm [51]. In FM partitioning, an initial (possibly invalid) partitioning p_0 is chosen. A number of iterations of the outer FM loop generate partitionings p_n, $n = 0, 1, 2, \ldots$, where p_{i-1} is improved to p_i in the ith pass of the loop. A single loop iteration consists of the formation of a list of the vertices in V. The list is ordered by the *gain* $g(v)$; $g(v)$ is the net improvement in cut if vertex v is moved to the opposite partition. The list is traversed in order, with the gains updated after each move. At the end of the traversal, the point in the list at which the minimum cut was reached is determined, and the moves after that point are reversed. The remaining moves constitute the improvement of p_{i-1} to p_i; if the cuts $c(p_i)$ and $c(p_{i-1})$ are equal, then there is no improvement, and FM stops.

FM is thus an iterative-improvement-based heuristic method; it is known that the quality of FM partitionings degrades somewhat with an increase in hypergraph size, mainly due to

the inability of the FM algorithm to reach a large part of the solution space [52]. Thus, multi-level FM techniques have been proposed that are themselves recursive [52–54].

Alternatives to FM partitioning, such as partitioning by iterative deletion, have also been proposed [55]. In this algorithm, a *redundant partitioning* is formulated in which each vertex is initially assigned to both partitions. From alternating partitions, a vertex is then successively selected and deleted. The choice of vertex is again motivated by a desire to minimize the cut while maintaining balance constraints. Iterative deletion stops when each vertex is assigned to exactly one partition.

Hierarchical placement proceeds by partitioning the design over the available placement area. For each partitioning, the available area is allocated to the partitions according to the weights W_i. The total cell area, represented as a rectangular block of cell rows, may thus be split in two ways: horizontally or vertically (three ways are possible in three dimensions, as we will address in Section 2.7.1). The choice of direction is typically motivated by the aspect ratio of the block. The result of a block partitioning is thus two sub-blocks with portions of the block's cells allocated to each of the sub-blocks. Each of these sub-blocks then also undergoes partitioning.

2.4.6 Detailed Placement Algorithms

If the size of the sub-blocks (in terms of cell count) falls below a certain threshold, it may become more effective to use optimal partitioning codes [56]. The use of techniques such as dynamic programming allows for an efficient exploration of the entire solution space. Similarly, for the end case, when the precise placement of the individual cells must be determined, optimal placement may be considered if the case size is sufficiently small. However, exhaustive search must be ordered using techniques such as Gray-code enumeration and pruned using methods such as branch-and-bound since the solution space is of size $O(2^n)$ for partitioning and $O(n!)$ for placement.

A typical placement implementation may use multi-level FM partitioning as a global placement algorithm together with optimal partitioners and placers for the detailed stage [47]. Alternatively, the detailed stage may use the above simulated-annealing or quadratic algorithms combined with a slot-assignment legalization step.

In detailed placement for 3-D ICs, the opportunity exists to explore routing trade-offs involving the inter-wafer vias. Depending on the routing strategy, modifications to the

wire-length estimation technique can be made.

2.5 Routing

The task of routing, much like placement, is typically divided into global and detailed stages. The key issue is one of concurrency: while it is possible to route all the wires sequentially, routing a given wire completely before proceeding to the next, this strategy is suboptimal since the routing of any given net affects the available options for routing of subsequent nets. Thus, the global stage is utilized for route planning, with a view toward optimizing various metrics such as congestion or cycle time. The detailed stage is used for determining the specific paths for the nets using the guidance of the global routes [57].

To first order, routing for three-dimensional integrated circuits may be seen as an extension of traditional multi-level routing techniques. Specifically, current algorithms can perform over-the-cell (OTC) routing using six or more metal levels, of which two are reserved for intra-cell routing and the remainder for inter-cell routing. In a 3-D integration technology with six metal levels for each of n device layers, the problem may be thought of to some extent as a $4n$-level OTC routing.

However, the use of inter-layer vias imposes additional constraints. Since inter-layer vias pass through the device layer, these vias can be permitted in a limited number of regions. Furthermore, as shown in Chapter 1, in some technologies the vias are formed after bonding, which implies that they pass not only through the device layer, but also through all 2-D metallization layers. Thus, these vias present obstacles to within-wafer routing as well.

It is clear, then, that inter-wafer vias must be handled at the earliest possible stage of the physical design process. In our 3-D placer, we detail strategies for allocating routing area for these vias. In routing, we must tackle this problem during the global stage. If solved then, detailed routing may be performed by conventional means.

Since inter-wafer vias present a unique obstacle, it is beneficial to consider routing strategies that allow us to minimize their use. The hierarchical method of Burstein and Pelavin [58] is one such method. The trade-off for utilizing a hierarchical method is that it is more difficult to optimize the performance of critical wires. Thus, we also consider the more traditional sequential approach of maze running.

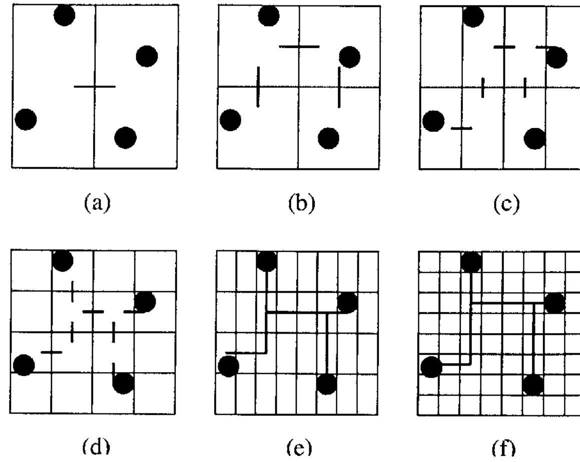


Figure 2-5: Single-net example of the hierarchical routing procedure. Routing proceeds from stage (a) to (f) by recursive partitioning.

2.5.1 Hierarchical Approach

In a hierarchical global router, the routing substrate (which consists of the wiring surface above the placed cells) is recursively bisected into routing subregions. Each side of each region has an associated capacity, which limits the number of wires that may enter the region through that side. Wires within a region may either be fully contained by the region or terminate at a *pin* on one or more sides of the region; initially, all wires are contained within the routing region. At each partitioning step, the existing pins on the sides of the routing region must be allocated to one of the two subregions. Those wires that are fully contained within the region must be allocated to one or both subregions. The remaining wires connect cells on both sides of the partition line; these are cut by the partition, and for each, a pin is inserted into the side between subregions. The manner in which existing pins are allocated to subregions dictates the quality of the overall routing. When complete, the resulting regions may be fed to a detailed router as formulations of channel or switchbox routing problems. Figure 2-5 shows a sample routing for a single net.

For the purpose of allocating inter-layer vias, we may proceed in two directions. If, as is likely the case, inter-layer vias are an expensive commodity, we may choose to use the first partitioning step to split the routing substrate into separate device layers. On the other hand, if optimal wire length is desired, it is best to use an aspect-ratio based sequence similar to what we will detail for 3-D placement in Section 2.7.1.

2.5.2 Global Maze Router

The maze routing approach, in constrast, considers the nets sequentially [59–61]. That is, the routing substrate is first divided into regions. Each region in this *global routing grid* is then pre-assigned a routing capacity – indicating the number of wires that it may contain – and a congestion value – a measure of how many pre-routed wires and other routing obstacles occupy the region. The unrouted nets are ordered according to any of several criteria (e.g. longest first or shortest first, as determined by half-perimeter length estimation). Each net in the list is routed by connecting the terminals on the net in sequence. A pair of terminals is connected by starting at one terminal and using a graph-based search to find an optimal path from that terminal to the other, where the optimization considers both the routed wire length and the congestion values for the regions along the chosen path.

These algorithms vary in time and search-space complexity depending on the implementation. Initial versions used breadth-first search [59]; improvements include the use of a detour number [60] or general A^* search. As before, when global routing is complete, the regions may be fed to a detailed router.

For routing of 3-D ICs, the primary algorithmic choice is in the ordering of nets. In 3-D ICs, it is likely to be most efficient to route multi-wafer nets first, as the required inter-wafer vias will present obstacles to routing other nets and will be more difficult to route in congested areas.

2.6 Layout

When the routing stage is complete, the resulting design is said to be laid out. A designer who chooses to forgo automated placement and routing may lay out the design by hand. In either case, a layout editor that permits manual entry of 3-D IC designs, as well as analysis and simulation of those designs, is needed.

The required functionality may be delineated as follows:

- *design management* – the layout information must be captured so that it is clear that the various device layers of a 3-D integrated circuit are associated. Concurrently, the individual device-layer designs should also be reuseable as single-layer (i.e. conventional) designs.

- *user interface* – the design methodology must not differ substantially from what is typical for conventional ICs. The extra dimensionality must be handled in a way that does not require an unwieldy use of the computer display.

- *layout vs. schematic (LVS)* – the interface must be able to provide the designer with topology information (i.e. connectivity and hierarchy) that spans all the device layers of the design, such that the functional accuracy of the circuit may be visually inspected.

- *design-rule checking (DRC)* – in addition to conventional design rules, the editor must support the implementation of tests for 3-D-specifc rules such as those involving alignment.

- *extraction* – the editor must be able to obtain topological information for the 3-D circuit, including parasitic components, from the layout.

Prior work on the development of transistor-level layouts for 3-D ICs has focused on methodology. For example, the method of S. Alam [62] includes the novel use of conventional features in the popular open-source layout editor Magic [63]. By combining a scheme for the association of design files in directories, a file-interchange system for the communication of inter-wafer interconnect information between device layers, and an augmented technology definition file that includes inter-wafer vias, this methodology makes good use of existing tools. However, it does not provide all of the functionality desired above.

Having identified the design flow, tools, and algorithms necessary for the development of three-dimensional integrated circuits, we describe our implementation of these tools in Sections 2.7 and 2.8.

2.7 PR3D: The Placement and Routing Tool

PR3D is the first major design tool we have developed to address the above issues and gaps in the flow for 3-D integrated circuits. It is a CAD tool for standard-cell circuits that covers the placement and global routing stages. In the following sections, we will describe the design of PR3D and the algorithmic choices underlying this design; a discussion of the use of PR3D for analysis of 3-D integrated circuits takes place in the next several chapters.

```
Algorithm 3DPLACE
  calls PARTITIONING
  calls PLACE_SINGLE_ROW
begin
  blocklist <- top level block
  newblocklist <- new list
  finishedblocks <- new list
  while (blocklist is not empty)
    while (blocklist is not empty)
    begin
      remove first block from blocklist
      if (block is a single row of six or fewer cells)
        PLACE_SINGLE_ROW
        add row to finishedblocks
      else
        choose partition direction for block
        if (direction is vertical)
          do a rough (20% tolerance) PARTITIONING to find the midpoint
        else if (direction is horizontal)
          do a rough (20% tolerance) PARTITIONING to find the middle row
        else (direction is parallel to wafers)
          set tolerance to make even split of wafers
            (e.g. 33% for 3 wafers, 20% for eight wafers)
          do a rough PARTITIONING to find the middle wafer
        endif
        do a refined (2% tolerance) partitioning around the mid point
        split block into two child blocks with area ratio equal to
          the area ratio of the cell partitioning
        add child blocks to newblocklist
      endif
    end while
    blocklist <- newblocklist
    newblocklist <- new list
  end while
end 3DPLACE
```

Table 2.1: Algorithm 3DPLACE for multi-wafer placement using min-cut partitioning.

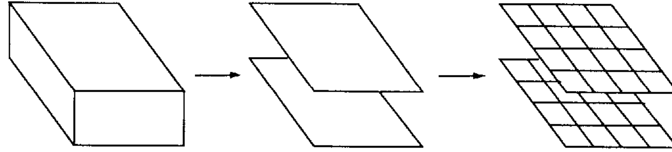


Figure 2-6: Partitioning strategy where plane assignment is done first in order to minimize the number of inter-plane vias.

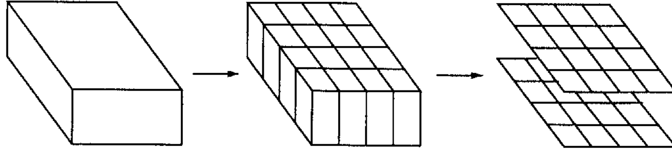


Figure 2-7: Partitioning strategy where plane assignment is done by considering aspect ratio in order to minimize total wire length.

2.7.1 3-D Standard-Cell Placement Algorithm

For the reasons outlined in Section 2.4, we have implemented PR3D as a partitioning-driven placement tool. Thus, our placement framework consists of the embedding of a hypergraph representation of a netlist into a rectangular block that represents the available die area. We assume that the dimensions of the block (number of rows, width of each row) are fixed *a priori* (i.e. a fixed-die context). For 3-D integration, given a set number of device layers (specified at run-time by the user), we adjust the number of rows and widths of each row (prior to execution) such that the total area available for placement remains the same as in 2-D and the aspect ratio for each device layer is the same as in 2-D.

We proceed by recursively partitioning the block roughly into halves, assigning nodes to each partition such that the capacity of each partition is not exceeded and the number of hyperedges spanning both partitions is minimized. Each partitioning step is permitted a tolerance varying from 2% to 20% depending on the discreteness of the partition. Partitioning into wafers or parallel to rows, for example, must be done very precisely since the resulting partition sizes must be integral numbers of rows or wafers, but when partitioning perpendicular to rows, a higher tolerance will yield a better partitioning.

We note that min-cut partitioning along the 3rd dimension is equivalent to minimizing the number of inter-layer vias. Thus, in cases where such vias are costly (due to capacitance, pitch, or fabrication expense), we may trade off increased total wire length for fewer inter-plane vias by varying the point at which the design is partitioned into planes. For example, we may choose to partition into planes first (as shown in Figure 2-6), or we may leave plane

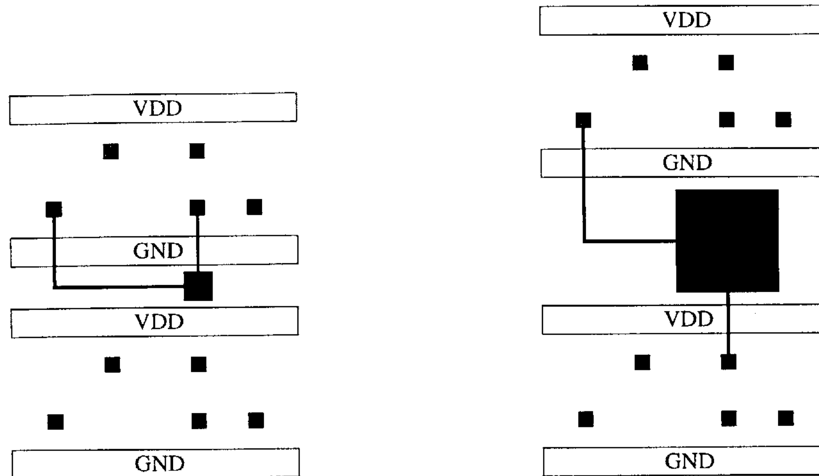


Figure 2-8: For small inter-wafer via sizes, we permit same-row interconnects to be split among multiple wafers. For large inter-wafer via sizes, we partition into wafers before reaching the single-row block size.

assignment until the detailed placement stage (Figure 2-7). We find that the optimal wire length is obtained by using aspect ratio to determine the cut sequence – that is, a given partition is bisected perpendicular to the longest dimension of the partition. (For purposes of comparison, the length of the third dimension is scaled by the cost of inter-layer vias.) The user specifies at run-time whether to minimize total wire length or number of inter-layer vias, as well as the cost of these vias.

Figure 2-8 shows qualitatively how the partitioning scheme should vary with the inter-layer via size. Copper wafer-bonding technologies [21, 24] are reflected in the left half of the figure, whereas dielectric-bonding technologies [7, 23] may be represented by the right half. In the former, same-row wiring in 2-D layouts may potentially be implemented more effectively by partitioning the cells on that wire over adjacent wafers. In the latter, this is less likely to be true; thus, partitioning into wafers should be done before the single-row block size is reached. (It should be noted that both figures are considered to be exhibiting a high-density interconnect – the spectrum of inter-wafer feature sizes is large enough that some technologies offer inter-wafer interconnects an order of magnitude larger than those in the right half of Figure 2-8.)

In partitioning a given block, we account for the presence of external nets by using a terminal propagation scheme based on that of Dunlop and Kernighan [48]. We extend this scheme to 3-D by expanding the dummy terminals used for propagation to include nodes locked to planes above or below the partitioning point. At very detailed levels, we use

branch-and-bound partitioning and placement [56]. Finally, wire lengths are determined using the half-perimeter metric, which in 3-D ICs is the sum of the length, width, and height of the bounding box containing all terminals of a given net.

Pseudocode for the 3-D placement algorithm is given in Table 2.1.

2.7.2 3-D Global Routing

Global-routing algorithms may generally be categorized as sequential approaches (such as maze routing) or concurrent approaches [57]. We have chosen to implement two global routers for 3-D integration: a concurrent (hierarchical) router [58] and a traditional maze router based on the A* algorithm.

Since modern technologies offer many levels of metal interconnect, we adopt an over-the-cell routing strategy. We assume that inter-cell wires may be routed without restriction on the upper levels of metal. The lower levels are reserved for intra-cell wiring, as well as power, ground, and other critical wires such as the clock tree. This uniformity in the routing substrate permits us to investigate hierarchical approaches based on concurrent methods.

Our 3-D global router considers a routing region to be a set of aligned, congruent 2-D routing regions on one or more adjacent wafers. Wires may enter or exit the region through any of the sides of the 2-D regions, as well as the top and bottom of the set. The 3-D router must therefore determine the location and quantity of inter-wafer vias in addition to routing the wires on each wafer. In 2-D ICs, it was assumed that cells would not interfere with the routing area; with inter-wafer vias this may not be the case, since these vias must punch through the device layer to contact metal. However, given a strategy for placing the inter-wafer vias, the remaining wire routing is a conventional problem.

There are three candidate strategies for placing these inter-wafer vias. The first is to allocate 3-D feed-through cells within the rows. For each wire to be routed between two wafers, a pair of matching cells is inserted, one in each wafer. This problem is not unlike that of inserting repeaters in long wires. However, relative to the repeater-insertion problem, a far greater number of wires will require 3-D via insertion. Furthermore, unlike repeater insertion, it is harder to predict in advance the number of wires that will require inter-wafer vias (as we will discuss in Chapter 3). As a result, by the time this information is known, it is not possible to allocate enough area for the vias without disturbing the quality of the placement.

Another strategy for inter-wafer via routing is to route wires directly to the source or drain of a driver transistor on the upper wafer. This avoids the area penalty associated with having to punch through the device layer on the upper wafer. However, there exist difficulties with this approach. First is that the technology for direct source or drain backside contact must be developed. Second, this approach does not cover situations where the upper wafer contains only loads (i.e. transistor gates) or where the wire connects cells on non-adjacent wafers (e.g. a cell on wafer 1 and a cell on wafer 3).

We therefore consider a third strategy: inter-row via placement. By separating the cell rows, we may create a pre-allocated space for 3-D vias. If the separation is small, the impact on placement quality is minimal. We thus limit the total capacity for inter-wafer vias to a single row's worth of vias per row of cells on a wafer (e.g. if a given wafer has ten rows of cells, and 50 vias can fit side-by-side within the width of a row, then the wafer has a inter-wafer via capacity of 500). With this capacity computed, 3-D global routing proceeds using either of the above algorithms.

Once completed, the results of global routing are computed as the sum over all routing regions of the half-perimeter wire lengths of the wires contained within each region. This measurement should more closely reflect the final aggregate wire length.

2.7.3 Comparison of PR3D with Other Tools

Having described our implementation of placement and routing tools for three-dimensional integration, we must justify this effort in light of the existence of similar tools. Specifically, the effort may be wasted if performance analyses can be made with existing tools.

Certain prior works are largely theoretical in nature and therefore not feasible for use in considering large circuits. The placement engine due to T. Tanprasert [64], for example, uses a nonlinear-programming formulation that does not scale well and has only been tested on circuits with a small number of modules (e.g. 15).

Other, more scalable competing placement tools exist. Y. Deng *et al.* has developed an extension to the open-source 2-D placement tool Capo [39]. This tool produces two-wafer implementations only and considers the wafer-partitioning-first approach outlined above in Figure 2-6, without regard for optimal-wire-length approaches.

In contrast, the 3-D VLSI tool Gravity, developed by S. Obenaus *et al.* [65,66], analyzes placements in a substrate of dimension $2+\epsilon$ (i.e. where cells may be localized to an $n \times n \times n^{\epsilon}$

grid). The fundamental technology assumption (which is different but not invalid) is that the number of wafers used will scale with the size of the circuit. For this reason, both direct comparison with the performance of Gravity and use of Gravity for wafer-by-wafer analysis of performance improvements using 3-D integration are made difficult.

Additionally, in all three cases, the tools have been designed strictly for use with benchmark circuits. In keeping with our stated goal of being able to analyze actual circuit performance and the desire to produce tape-out quality layout for eventual fabrication, we have designed PR3D with capabilities that exceed those of the other 3-D place-and-route tools in existence. Specific usage information for PR3D is given in Appendix A.

2.8 3-D Magic: The Layout Editor

To achieve the desired functionality stated in Sections 2.6 and 2.7, it is necessary to develop an actual layout editor for 3-D ICs rather than a methodology that can be used with an existing conventional tool. However, rather than develop a layout editor from scratch, we find it useful to add functionality to an existing editor. This has the dual benefits of being a less complex undertaking and producing a final product whose use is familiar to existing users.

The conventional layout editor Magic [63] is a versatile tool that is popular in academia as well as industry since both the binary executable and source code are free and readily available. We therefore implement our layout editor for 3-D ICs as an extension to this tool and call it 3-D Magic.[4]

In the following sections, we detail considerations regarding the design of the interface, internal data representations, and technology-specific modules such as LVS and DRC verification and parasitic extraction.

2.8.1 User Interface Design

The conventional tool Magic is a paint-based design tool; the user draws the transistor geometry much as one would in a painting program, and Magic converts this geometry to mask layout.

[4]The methodology of S. Alam *et al.* has also been named 3-D Magic. Our work is distinct from theirs and consists of software that can be used to implement that methodology, as well as others, in a more automated fashion. While the methodology predates our work, both are unfortunately called 3-D Magic in the literature.

In 3-D Magic, we adopt a previously-developed user interface design in which several key issues were delineated for a technology-specific circuit editor [67]. We must also consider these issues for the technology-flexible design tool we wish to produce. For example, it is desired that the tool be able to handle an arbitrary number of device layers in a single 3-D design. This may be addressed in one of two ways: the entire design may be managed as a single unit with some number of co-dependent views, or the design may be handled as a collection of individual device layers with their corresponding traditional views. Since the easiest transition for a designer is to use conventional 2-D circuit views, it is only natural that the design and user interface be partitioned into individual wafers.

Thus, we seek to implement a 3-D IC layout as a collection of designs for the individual wafers. The wafer designs are associated within 3-D Magic by the issuing of a command to the interface that tells the tool which other design is mated to the actively-edited design, and to which side the bond occurs. Once bonded, the 3-D IC's inter-wafer interconnects are represented in the interface as conventional vias; however, we extend Magic to automate the display of the designed via connectivity over all wafers spanned by the via.

Figure 2-9 shows two windows in which a two-wafer design is being laid out. The row-aligned square pads represent inter-wafer vias.[5] In 3-D Magic, once two wafers are bonded into a 3-D IC design, the user interface automates the display of inter-wafer vias across all relevant wafers. Each design window shares a global coordinate system used to align inter-wafer vias across the design. When a designer places an inter-wafer via on a wafer of a bonded pair, the interface indicates a hint on the corresponding wafer. The designer then paints the corresponding metallization on that wafer.

2.8.2 Circuit Issues

In addition to determining how to visualize and manage design information for a 3-D IC, we must also provide specific circuit functionalities. We will address LVS, DRC, and parasitic extraction issues here.

Layout-versus-schematic information is provided to the designer through an extension to the Magic "selection" feature. In Magic, the user may select portions of the layout and ask the interface to identify the electrical node (i.e. all connected layout) to which the

[5] The alignment is a product of the standard-cell design methodology; in general custom circuits, these vias may be placed arbitrarily.

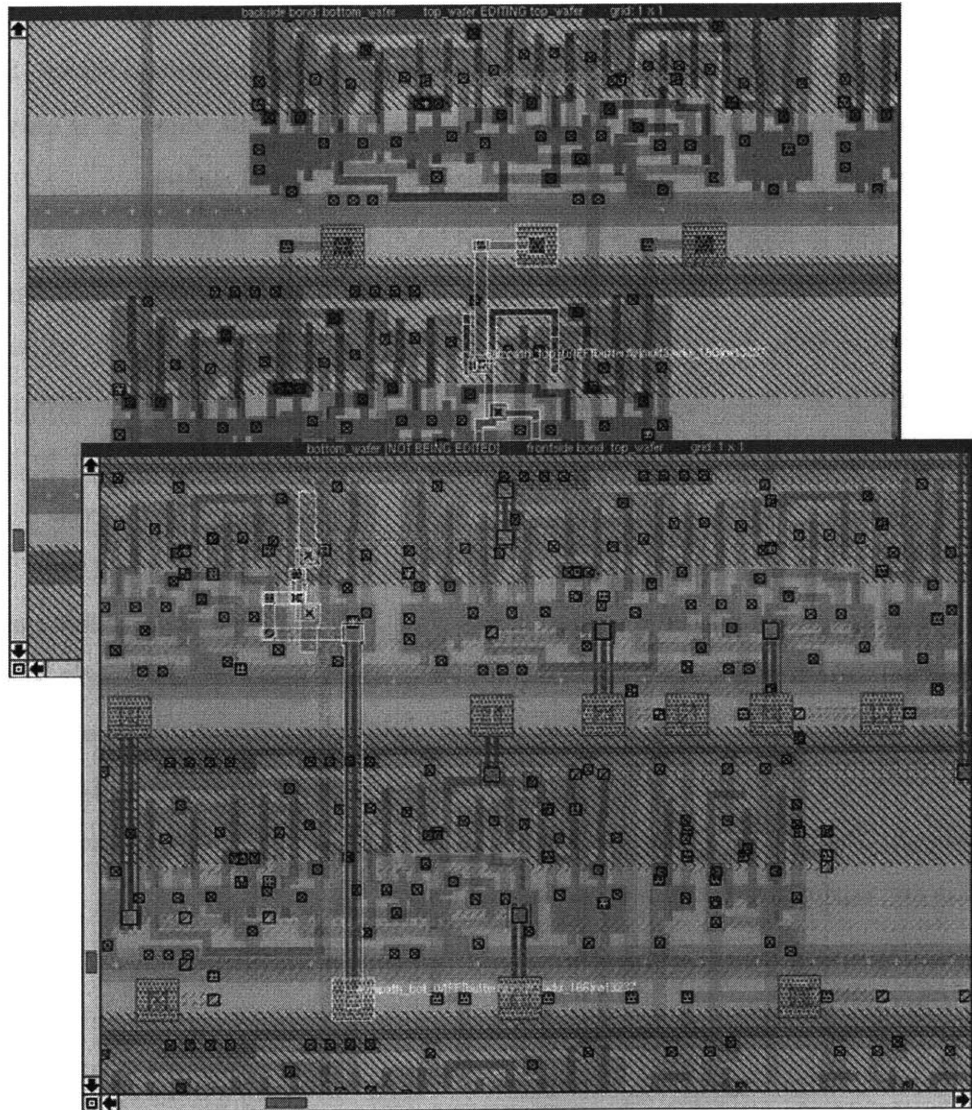


Figure 2-9: Screen shot of 3-D Magic exhibiting a two-wafer circuit layout.

selected portion belongs. If empty space is selected, the interface will identify the hierarchy of cells containing the selected point. For 3-D IC design, the ability to select electrical nodes that span multiple wafers is critical since these nodes are separated over multiple design windows. This feature is not available in methodology-only design flows that use conventional design tools. In Figure 2-9, the wire outlined in white is a selected electrical node that spans two wafers.

Three-dimensional integration also presents unique design-rule issues. The primary issue is one of alignment: as our system separates design information according to device layer, it is important to provide immediate feedback, should a designer place inter-layer contacts that do not align exactly.

Two features that are not handled by methodology-only flows are circuit connectivity and parasitic extraction. For parasitic analysis, one desired aspect is the capability for whole-circuit extraction: it is useful for the layout editor to be able to produce this without user intervention, rather than requiring the user to ensure that all inter-layer contact points carry the same electrical labels. Another aspect is the ability to determine parasitic interactions between the inter-layer interconnects and other structures in the circuit. In 3-D Magic we provide both parallel-plate capacitance extraction data (e.g. for cases in which the contact is formed by bonding of two copper pads) and a lumped-parameter interface. This lumped parameter interface allows the designer to substitute a parameterizable model for the inter-layer interconnect, so complicated structures such as solder-bump bonds can be laid out using via-type paint layers.

In the next section, we detail the extensions we make to Magic's internal data representation to support the above functionality.

2.8.3 Data Representation

In Magic, the internal representation of a single-wafer circuit is stored in a data structure called a `CellDef`. To extend a `CellDef` for integration of multiple wafers, we incorporate bonding information. Specifically, for each `CellDef`, we define a pointer `up` that links to the `CellDef` bonded to the front side (i.e. metallization side) and a pointer `down` that links to the back side (i.e. substrate). We also define two pointers, `prev` and `next`, with the condition that in a stack of bonded wafers, all `next` pointers point in the same direction and all `prev` pointers point in the reverse direction. The `prev` and `next` pointers thus may be

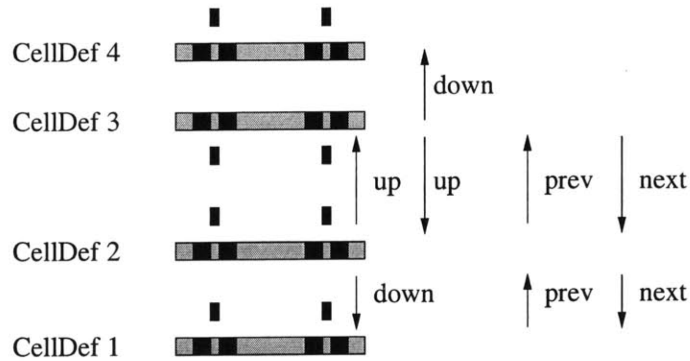


Figure 2-10: Bonded stack of CellDef structures with up and down pointers for front-side and back-side bonding contacts and prev and next pointers for stack traversal.

used to traverse the entire stack, while the up and down pointers are used for bond-specific actions such as the graphical rendering of inter-wafer vias. Figure 2-10 depicts the CellDef setup.

Two CellDef structures are bonded by issuing a :bond instruction to 3-D Magic's command module. The command-line arguments specify whether the bond is *flipped* (i.e. face-to-face) or *notflipped* (i.e. face-to-back). The up and down pointers are set accordingly. If either or both structures is part of a pre-existing 3-D stack, the prev and next pointers are aligned such that they point in the same physical direction for both stacks. In case of user error or reconsideration, the :unbond command is also provided.

Specific information concerning the 3-D integration technology is provided via extensions to Magic's technology file format. This information is supplied in three sections. The first two are the extract and drc sections. In the extract section, parasitic coupling information for 3-D interconnects is provided in the same manner as for conventional metallization. The drc section incorporates a new rule, exact_overlap_3D, which specifies that the listed contacts must overlap exactly if any overlap exists. The third section, contact3D, is new: it specifies the inter-wafer contacts and the side (e.g. front side or back side) to which these contacts connect.

Electrical-node selection has been implemented by redesigning the architecture of Magic's select module. Conventional Magic implements a buffer called SelectDef into which the edit cell can be copied and manipulated. In 3-D Magic, a tree of SelectDef cells, indexed by cell name, is managed. The selection operations have been modified to traverse 3-D-bonded CellDef trees.

Similarly, the extract module has been re-architected to incorporate the spanning of

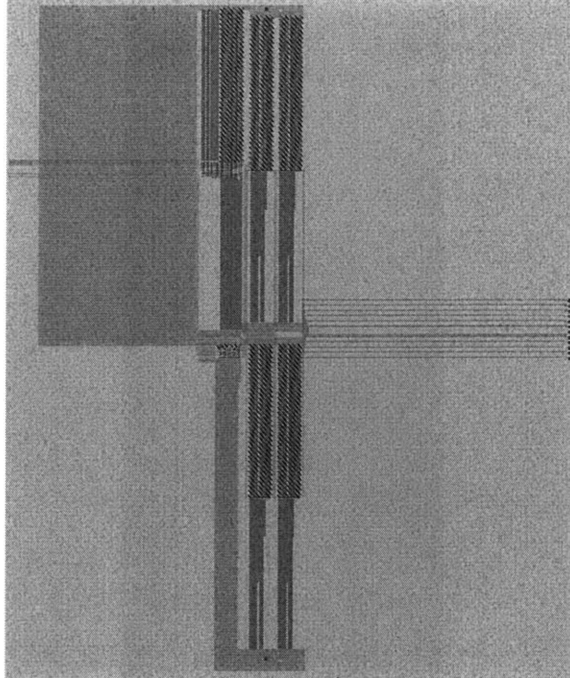


Figure 2-11: Bottom wafer of a two-wafer class-E amplifier designed by Wei-Han Huang and Vivian Lei.

`CellDef` trees. The extraction of a 3-D bonded stack, as opposed to the extraction of a single wafer in the stack, may be executed by issuing the `:extract stack` command.

2.8.4 Sample Layouts Using 3-D Magic

3-D Magic has been used to design a number of circuits. We present two of them here to illustrate the capabilities of the software. Complete usage information for 3-D Magic is provided in Appendix A.

Class-E Amplifier

Students Wei-Han Huang and Vivian Lei designed a CMOS class-E power amplifier with multiple tuning frequencies [68]. This design was laid out in a hypothetical 0.25 μm 3-D process with five metal layers. Figure 2-11 shows the layout of the bottom wafer, containing the analog amplifier circuitry for 13.5 MHz and 1.9 GHz implementations. The top-wafer layout, shown in Figure 2-12, contains digital and analog control circuitry. The per-wafer chip size is 377 μm × 444 μm.

The power efficiency for the two layouts was evaluated. Figure 2-13 shows that a two-wafer implementation improves the efficiency of the 1.9 GHz amplifier from 20% to 30%.

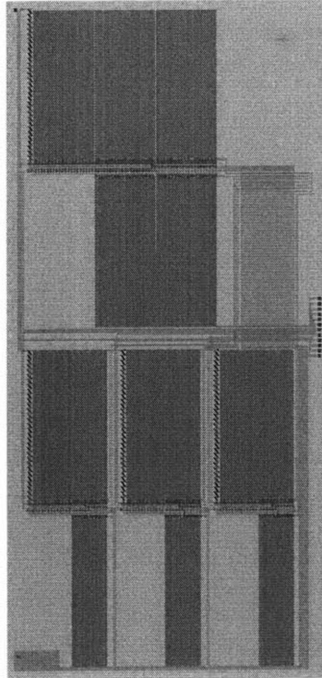


Figure 2-12: Top wafer of a two-wafer class-E amplifier designed by Wei-Han Huang and Vivian Lei.

Additionally, crosstalk on the long control lines in the digital selector subcircuit on the top wafer was evaluated. Figure 2-14 shows that this crosstalk can be reduced significantly.

Four-Bit Analog-to-Digital Converter

Students Elizabeth Basha, Katie Butler, and Patrick Griffin designed a four-bit analog-to-digital converter (ADC) [69]. This ADC was designed and laid out in a hypothetical 0.25 μm 3-D process with three metal layers. Figure 2-15 shows the block architecture for this ADC, and Figure 2-16 exhibits the final layout of the two-wafer design. The per-wafer chip size is 144 μm × 180 μm.

The design was compared to a reference single-wafer design to determine the substrate noise characteristics. A heavily-doped substrate model was used; the students computed the ratio of signal to noise and distortion (SNDR) and thus the effective number of bits (ENOB) for the ADC. Figure 2-17 shows the SNDR for both layouts, and Table 2.2 gives the ENOB.

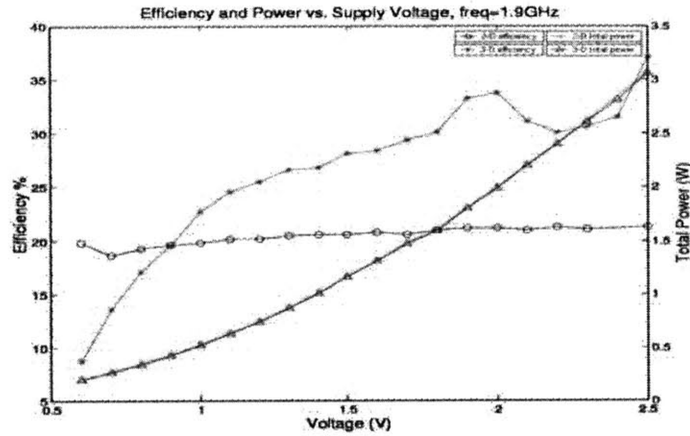


Figure 2-13: Power efficiency of the 1.9 GHz amplifier in 2-D (○) and 3-D (∗) implementations. Total power is given in third curve (Δ).

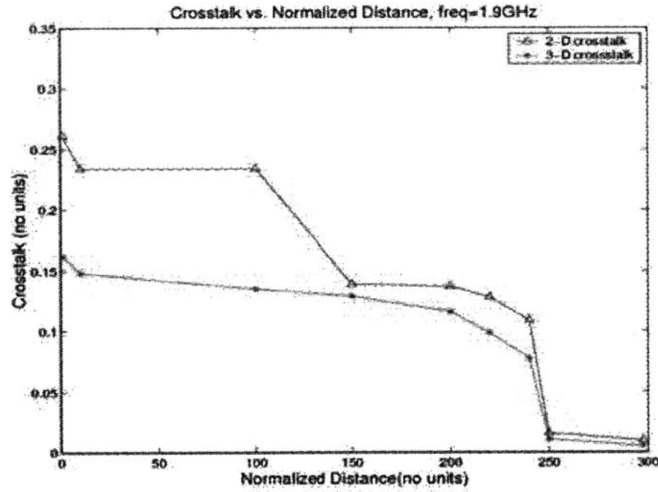


Figure 2-14: Crosstalk on adjacent multiplexer lines in the selector subcircuit of the 1.9 GHz amplifier, in 2-D (Δ) and 3-D (∗) cases, as a function of separation distance.

	signal range			
	0.25 V	0.5 V	1 V	2 V
2-D	2.7290	3.5512	3.8737	3.8950
3-D	2.9634	3.7797	3.9652	3.9599

Table 2.2: Effective number of bits (ENOB) for 2-D and 3-D implementations of the ADC.

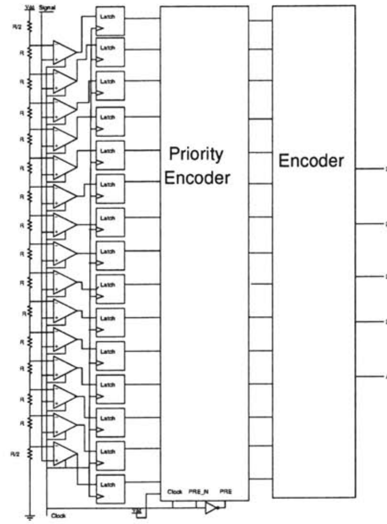


Figure 2-15: Block diagram for a four-bit ADC designed by Elizabeth Basha, Katie Butler, and Patrick Griffin.

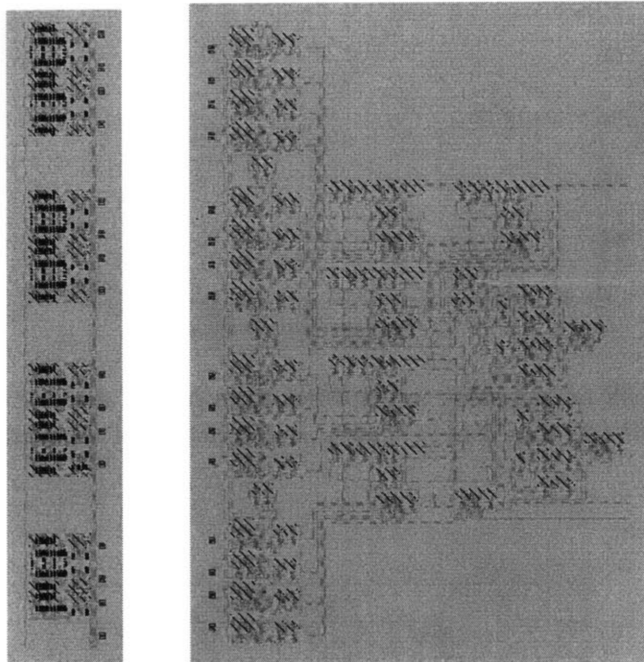


Figure 2-16: Top wafer (left) and bottom wafer (right) of the two-wafer ADC designed by Elizabeth Basha, Katie Butler, and Patrick Griffin.

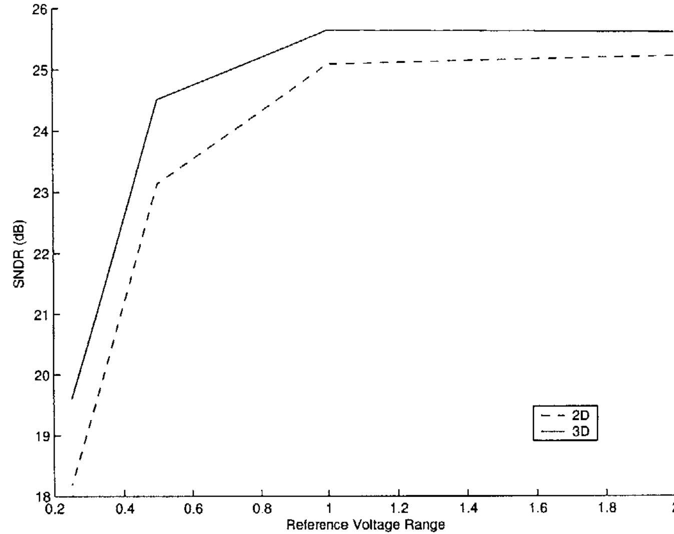


Figure 2-17: Signal-to-noise-and-distortion ratio (SNDR) for 2-D and 3-D implementations of the ADC.

2.9 Summary

In this chapter, we described our design flow for 3-D integrated circuits. The design of this flow and the tools used therein was motivated by a consideration of all stages of the conventional 2-D design flow. In our review of these stages, we determined when both awareness of 3-D integration technology was beneficial or necessary and when the requirement for 3-D integration might motivate algorithmic choices.

As a result, we have developed two CAD tools. PR3D is a design tool for the placement and global routing of 3-D standard-cell circuits. It is the first tool for such circuits that has been developed from scratch and can produce tape-out information suitable for fabrication.

3-D Magic is a layout editor for 3-D ICs. It is the first tool usable for mask design and verification for 3-D ICs of arbitrary technology. With 3-D Magic, it is possible to utilize the features of a 3-D integration technology by means that are familiar to designers of conventional circuits. Tools such as LVS, DRC, and parasitic extraction have been extended to treat 3-D ICs in the same manner that conventional ICs are treated. In particular, the versatility and usefulness of this tool was demonstrated by two layout-based case studies of circuit performance in 2-D and 3-D integration.

In the following chapters, we put these tools to use. By placing and routing various circuits using PR3D, we determine the extent to which three-dimensional integration can be beneficial.

Chapter 3

Wire-Length Performance of 3-D Integrated Circuits

3.1 Previous Work on 3-D IC Analysis

In anticipation of the development of functional three-dimensional integration technologies, several research endeavors were undertaken to predict the utility of 3-D integration for various types of circuits [8, 30, 70–75]. The underlying approach in all such endeavors has been the same: Given some basic technology assumptions, a mathematical model of a class of circuit networks is formed that can be used to compare the performance of 2-D and 3-D layouts of the various circuits. For example, a digital circuit consisting of a set of logic gates and associated wires may be modeled as a graph, in which graph nodes correspond to gates and graph edges correspond to interconnects.[1] A mapping of the graph into a circuit substrate is then determined. The graph nodes are assigned unit dimensions, from which area and wire-length estimations can be made.

Initial research focused on the embedding of classes of circuits into general graph topologies, which could be optimized for 2-D or 3-D integration [70, 72, 73]. For example, N-node 2-D grids, 3-D grids, binary trees, and fat trees were considered (Figure 3.1 shows a typical fat-tree). The substrate was modeled as a planar grid (for 2-D) or a 3-D grid with infinite extent in all dimensions. Optimal embeddings of the above networks into these two substate models were then determined. Using this methodology, several regular circuit

[1]Multi-terminal interconnects are typically modeled using collections of two-terminal edges (such as completely-connected or star-graph models), the details of which are not directly relevant to this discussion.

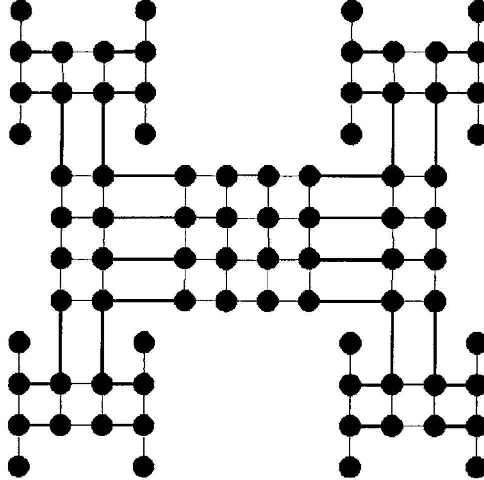


Figure 3-1: N-leaf planar fat-tree network exhibiting $O(\sqrt{N})$ bisection bandwidth.

topologies were shown to be improvable using 3-D integration. For example, an N-point Fast Fourier Transform (modeling a single element of the butterfly network as a graph node) can be implemented in area $O(N^{3/2})$ with the longest wire of length $O(N^{1/2})$ in the above 3-D technology. The same circuit requires $\Omega(N^2)$ area with the longest wire of length $\Omega(N/\log N)$ in two dimensions. A general N-node circuit requiring A area in two dimensions may be implemented in area $nA^{1/2}$ using three.

Obvious limitations exist in the modeling aspects of this approach. First, the 3-D technology is modeled as an infinite grid. However, many otherwise useful 3-D integration technologies are not arbitrarily scalable in the third dimension. This method does not have the capability to model a technology with a fixed number of device layers; specifically, since the model studies order-of-magnitude performance as a function of circuit size, it predicts zero improvement in cases in which the number of layers is fixed. The relevant performance increases due to 3-D integration are obtained in the constants hidden by order-of-magnitude analysis.

Second, the graph representation of circuits does not permit an accurate modeling of interconnect. Circuit wires are not specifically modeled in the mapping of a graph into a substrate; instead, a planar graph superset (such as the fat tree in Figure 3.1) is used, in which dummy nodes represent non-adjacent logic-gate connections. From a technology perspective, this model may be used to represent technologies with one or two metal routing layers (prevalent at the time that the model was introduced). Furthermore, in the case most favorable to interconnect, in which only leaf nodes are used for logic gates, the available

bisection bandwidth (number of wires or graph edges that intersect a bisection of a subgraph or subcircuit) grows as $O(N^{1/2})$, where N is the number of logic gates in the subcircuit.

However, Landman and Russo observed that in general, an empirical relationship exists between the number of logic components in a subcircuit and the number of I/O terminals needed to connect to the subcircuit that may be expressed as

$$T = kN^p. \tag{3.1}$$

This is known as *Rent's rule* [76], and the parameters k and p, known as the *Rent coefficient* and *Rent exponent* respectively, are properties unique to a given circuit. In a typical circuit, $1/2 < p < 1$, such that even the fat-tree topology is not sufficiently scalable. (This can in fact be seen as a condemnation of 2-D technologies with a fixed number of metal layers, since the scalability of such technologies is also $O(\sqrt{N})$. The *interconnect bottleneck* associated with large circuits in a 2-D technology is a consequence of this phenomenon.)

Thus, a large class of analytical models is based on Rent's rule [8, 30, 74, 75]. Rather than attempt to determine a physical mapping of a graph topology into a substrate, these models use *stochastic* (statistical or probabilistic) methods of determining the locations of logic gates and the distribution of wires. Given a circuit with a distribution of N logic gates over an area A, the number of wires of length l in the circuit is predicted by

$$f(l) = q(l)D(l), \tag{3.2}$$

where $D(l)$ is the number of valid two-terminal wire locations (a wire location is specified by the locations of the terminals) and $q(l)$ is the *occupancy distribution* (the probability that a location is occupied by a wire) [77]. For technology independence, values of l are typically given in units of gate pitches, where one gate pitch is the width of a (hypothetically square) single logic gate. The models cited above differ in the means used to determine $q(l)$ and $D(l)$, but all of them utilize Rent's rule in the derivation of $q(l)$. These models may be categorized as either hierarchical (where the algebraic form of $q(l)$ is dependent on l) or non-hierarchical.

Our goal is to determine the extent to which circuit performance can be improved by three-dimensional integration with as little *a priori* information as possible. Therefore, we seek to use as generic a model as is available; in particular, we avoid some hierarchical

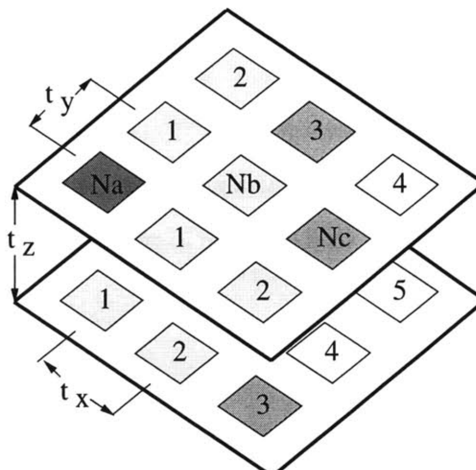


Figure 3-2: Schematic representation of the derivation of occupancy distribution: $N_a = 1$ is the logic gate in question, N_c is the number of target logic gates at Manhattan distance l gate pitches, and N_b is the number of logic gates in between. t_x, t_y, and t_z are the gate width, height, and inter-layer thickness, respectively, in micrometers. (Figure courtesy A. Rahman.)

approaches tailored for modeling the specific top-down algorithm used for actual circuit placement. In other words, while models may exist that more accurately predict the result of our 3-D placement engine, we do not wish to confine our analysis to the determination of what is possible within our placement tool.

Therefore, in the following sections, we examine a specific type of non-hierarchical model: the Rahman model for 2-D and 3-D integrated circuits. We adapt this model for use with standard-cell circuits, with the primary purpose of comparing the predictions of the model with measurements from circuits placed and routed using PR3D.

3.2 The Rahman Model

3.2.1 Derivation

The model of A. Rahman for three-dimensional integrated circuits [8] of which we give a derivation here is based on the model of J. Davis for conventional integrated circuits [78].

In this model, the wire-length distribution is given by

$$f_{3D}(l) = q_{3D}(l) M_{3D}(l). \tag{3.3}$$

The occupancy distribution, $q_{3D}(l)$, is computed from an interconnect distribution

$$I_{3D}(l) = \frac{T_{a\to c}}{N_c} = \frac{\alpha k}{N_c}[(N_a + N_b)^p + (N_b + N_c)^p - N_b{}^p - (N_a + N_b + N_c)^p], \tag{3.4}$$

where I_{3D} is the estimated number of interconnects between a pair of gates separated by distance l. This is determined by computing the total number $T_{a\to c}$ of length-l interconnects from a given gate a (i.e. $N_a = 1$) and dividing by the number of gates N_c at distance l from gate a. $T_{a\to c}$ is computed using Rent's rule, where N_b is the number of gates separating gate a from the N_c gates at distance l, and k, p, N_a, and N_c are defined as before; α, equal to $\frac{\text{f.o.}}{1+\text{f.o.}}$ where f.o. is the average fan-out, is used to avoid multiple-counting. Figure 3.2.1 shows how N_a, N_b, and N_c are enumerated.

Given thc total number of interconnects in a circuit, I_{tot}, the occupancy distribution may be computed as

$$q_{3D}(l) = \Gamma I_{3D}(l), \tag{3.5}$$

where Γ is a normalization constant such that

$$I_{tot} = \Gamma \sum_{l=0}^{l_{max,3D}} I_{3D}(l). \tag{3.6}$$

In the Rahman model, the number of pairs $M_{3D}(l)$ of gates separated by distance l is calculated as a summation of the Davis-model $M_{2D}(l)$ over the various device layers. For conventional integrated circuits, the Davis model specifies that

$$M_{2D}(l) = \begin{cases} \frac{1}{3}l^3 - l^2 l_{max} + \frac{1}{2}l_{max}^2 l & 1 \le l < l_{max}/2 \\ \frac{1}{3}(l_{max} - l)^3 & l_{max}/2 \le l < l_{max} \end{cases}, \tag{3.7}$$

where l_{max} is the maximum length of a 2-D (single-layer) wire in gate pitches. Given $M_{2D}(l)$, we may compute

$$M_{3D}(l) = \sum_{i=0}^{N_z-1} \beta_i M_{2D}(l - it_z)u(l - it_z), \tag{3.8}$$

where u is the unit step function, N_z is the number of device layers, t_z is the inter-layer thickness (expressed here in units of gate pitches), and β_i are constants that depend on the number of device layers and the range of inter-layer interconnects [8]. (For the purposes of

this analysis, we assume that there is no restriction on the number of device layers that an inter-layer interconnect may span.)

The wire-length distribution for a given circuit may thus be computed as

$$f_{3D}(l) = \Gamma I_{3D}(l) M_{3D}(l). \tag{3.9}$$

3.2.2 Adaptations for Standard-Cell Circuits

For a given circuit, $f_{3D}(l)$ (l in gate pitches) depends on the fan-out, Rent parameters k and p, number of layers N_z, and the physical extent of the circuit l_{max}. The value of l_{max} in gate pitches may be expressed as the square root of the number of gates on a single layer. Also, the normalization constant Γ depends on the total number of interconnects, I_{tot}. Finally, in order to express the distribution as a function of l in meters or micrometers, the dimensions of the individual gates must be known, as well as the thickness t_z of individual layers.

Rent Parameters

Rent parameters for a given circuit are traditionally determined by recursive partitioning of the circuit netlist. However, it has been shown that a similar version of the Rent parameters can be derived from placement. Furthermore, Rent parameters from placement are believed to reflect more accurately the distribution of wires and the quality of placement. Indeed, wire-length estimation for 2-D circuit placements using placement-based Rent parameters is more accurate [79].

There are two generally-accepted methods of computing the Rent parameters. In both, gate and terminal counts are computed for sub-modules of the circuit at various levels of hierarchy [76]. The Rent coefficient and Rent exponent that provide the best fit to the data may then be found simultaneously [79]. This method typically produces a Rent coefficient that differs from the average number of terminals per gate. Alternatively, the Rent coefficient may be computed through the use of its definition as the average number of terminals per gate; the Rent exponent that best fits the data is found using this Rent coefficient. We use the latter method, as the predictive model assumes this value for the Rent coefficient.

It is expected that the greater number of nearest neighbors available to transistors in 3-D integrated circuits will lead to a shift in the wire-length distribution towards local wires

and a reduced need for inter-partition interconnects at any fixed partition size. In other words, the Rent exponent derived from a 3-D placement of a given circuit is expected to be less than that derived from a 2-D placement: For a given circuit,

$$p_{partition} < p_{n+1} < p_n < p_1, \tag{3.10}$$

for modest values of n. ($p_{partition}$ denotes the Rent exponent derived from partitioning, and p_i is the Rent exponent derived from a placement using i device layers.)

As our goal is to evaluate the Rahman model as an *a priori* estimation tool for circuits for 3-D integration, we utilize the Rent parameters extracted from partitioning and 2-D placement.

Circuit Dimensions

In standard-cell circuits, the gates and low-level modules are synthesized as rectangular cells of a fixed height and variable width. The die area is specified *a priori* as a fixed number of rows with a fixed height and width and fixed inter-row spacing (i.e. a *fixed-die* context). For 3-D ICs, we preserve the fixed-die context by scaling the number and width of rows by the square root of the number of device layers to maintain constant area for cell placement.

To determine the gate count and gate pitch, we use the size of the narrowest cell as the unit gate. The gate count is equal to the total cell width divided by the width of the unit gate. The horizontal and vertical gate pitches are given by the width of the unit gate and the row-to-row pitch respectively.

The layer-to-layer thickness t_z is given by the technology. In a wafer-bonded circuit, for example, t_z may be as low as a few micrometers. If a solder-bump interconnect interface is used, the thickness may not increase appreciably, but the capacitance of inter-wafer interconnects may increase by as much as an order of magnitude. For MCM-V packages, the average die-to-die interconnect length scales with the die size. Thus, t_z may be determined as a function of electrical parameters (or other parameters) of the inter-layer interconnect, rather than strictly as the distance between device layers.

	QPlace	Dragon	Capo	our placer
ibm01-easy	0.59	0.58	0.56	0.56
ibm01-hard	0.59	0.56	0.56	0.55
ibm02-easy	1.59	1.54	1.55	1.56
ibm02-hard	1.57	1.44	1.52	1.59
ibm07-easy	3.79	3.55	3.73	3.63
ibm07-hard	3.66	3.32	3.60	3.59
ibm08-easy	3.97	3.66	3.94	3.96
ibm08-hard	3.78	3.41	3.77	3.83
ibm09-easy	3.45	3.10	3.18	3.20
ibm09-hard	3.25	3.07	3.23	3.16
ibm10-easy	6.47	6.00	6.26	6.16
ibm10-hard	6.28	5.97	6.35	6.12
ibm11-easy	5.15	4.78	4.99	4.96
ibm11-hard	4.97	4.55	4.99	4.81
ibm12-easy	9.31	8.54	8.65	8.85
ibm12-hard	8.53	8.46	8.35	8.30
dev. from avg.	+3.2%	-3.5%	+0.4%	-0.1%

Table 3.1: Performance of our placer and other state-of-the-art placers on the IBM-PLACE 2.0 circuit benchmark set. Wire lengths are in meters.

3.3 Analysis of 3-D ICs: Model vs. PR3D

3.3.1 Calibration

Having described the Rahman system-level interconnect model and adapted it for standard-cell circuits, we now evaluate our candidate 3-D technology using both the model and our 3-D IC placement and routing tool, PR3D. We first calibrate PR3D against some leading-edge placement tools for conventional ICs: Cadence® QPlace®, Dragon [80], and Capo MetaPlacer [47]. Table 3.1 shows the placement results for the four tools on the IBM-PLACE 2.0 benchmark set. The benchmark circuits and placement data for the external tools were obtained from Yang *et al.* [81]. We observe that PR3D is competitive with state-of-the-art placement tools for conventional standard-cell circuits.

3.3.2 Verification of the Rahman Model

The most straightforward way to ascertain the accuracy of the Rahman model is to make predictions for a set of circuits and then compare them to actual layout data. Using PR3D, we have placed and routed the eight largest circuits from the ISPD '98 benchmark suite [82] using one through five device layers for each circuit. Table 3.2 gives the relevant data for

circuit	# cells	# nets	Rent k	Rent p (1)	Rent p (2)	Rent p (3)
ibm11	68119	67016	3.48	0.662	0.753	0.692
ibm12	69026	67739	4.26	0.685	0.755	0.715
ibm13	81018	83806	3.67	0.677	0.764	0.665
ibm14	145492	143202	3.51	0.689	0.787	0.719
ibm15	157861	161196	3.99	0.669	0.766	0.667
ibm16	181633	181188	4.16	0.675	0.765	0.705
ibm17	182359	180684	4.55	0.694	0.759	0.725
ibm18	210051	200565	3.89	0.671	0.741	0.707

Table 3.2: Cells of the ISPD '98 benchmark suite used in this study.

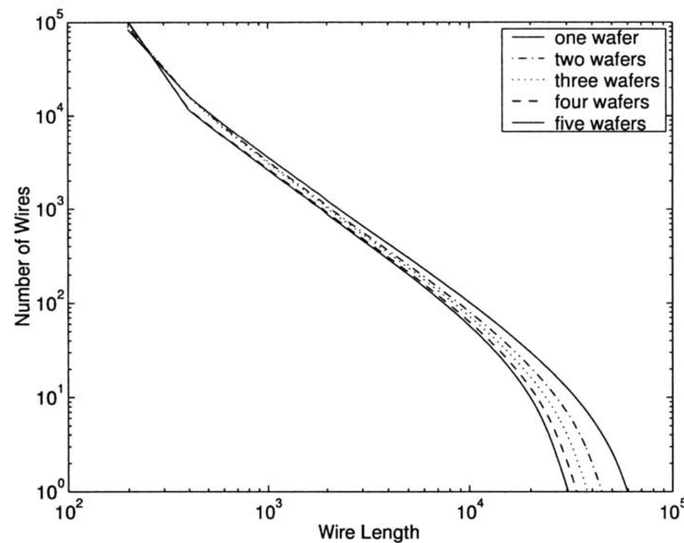


Figure 3-3: Predicted wire-length distribution for the ibm14 benchmark circuit with inter-layer pitch t_z of 1 micrometer.

these benchmark circuits. We provide the Rent exponent from partitioning and calculations using both placement-based methods described above. The value (1) is the Rent exponent from partitioning, value (2) is used in the model, and value (3) is computed using the same method as Yang *et al.* [79].

Figures 3-3 and 3-4 show that the basic mechanism underlying the Rahman wire-length predictions is sound: the predicted wire-length distribution as a function of number of device layers (Figure 3-3) matches well with the data from placement (Figure 3-4). Increasing the number of device layers does shift the distribution leftward, yielding more local wires and fewer global and semi-global wires. There are some discrepancies, however, that will be discussed.

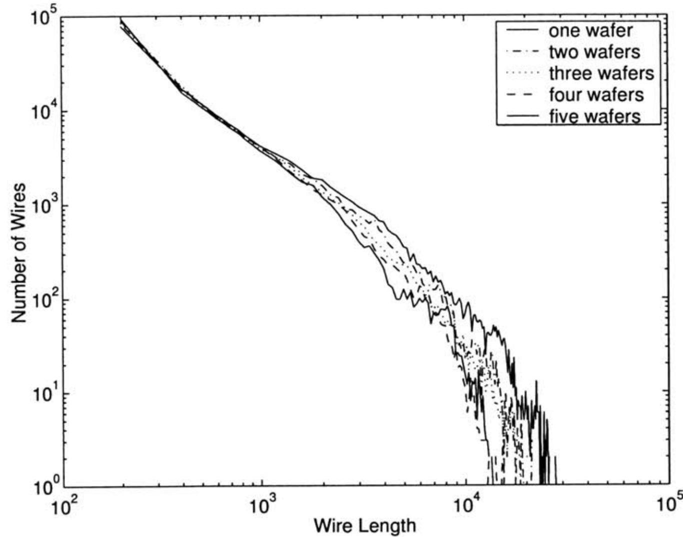


Figure 3-4: Placed wire-length distribution for the ibm14 benchmark circuit with inter-layer pitch t_z of 1 micrometer.

Figures 3-5 through 3-8 show data from the benchmark circuits. Figures 3-5 and 3-6 compare the total wire length of these circuits as predicted by the Rahman model and the total wire length of the circuits obtained by placement and routing, assuming an inter-layer pitch of 1 (i.e. assuming that an inter-layer via is equivalent to 1 micrometer of metal wire). In Figure 3-5, the wire lengths are normalized to the 2-D placement case and averaged over all eight circuits. Figure 3-6 shows the same data in which all wire-length curves are normalized to their 2-D cases, to demonstrate how the model predicts the percentage reduction in total circuit wire length as a function of number of device layers.

Similarly, Figures 3-7 and 3-8 compare the prediction of the Rahman model to the placement and routing outcomes for the same circuits, but where an inter-layer pitch of 250 is used (i.e. where an inter-layer via is equivalent to 250 micrometers of metal wire).

It can be seen that the Rahman model is fairly accurate in predicting how 3-D integration affects the wire lengths of circuits. We observe in Tables 3.3 and 3.4 that the wire-length predictions using the 2-D placement Rent exponent are within approximately 20% of the wire lengths obtained from placement as well as global routing. As expected, we find that the Rent exponents from 2-D placement more accurately reflect the character of the 3-D placements than the Rent exponents from partitioning, because of the use of terminal propagation in both 2-D and 3-D placement algorithms. The model and the placement and routing data thus show that 3-D integration provides useful benefits for digital circuits: a

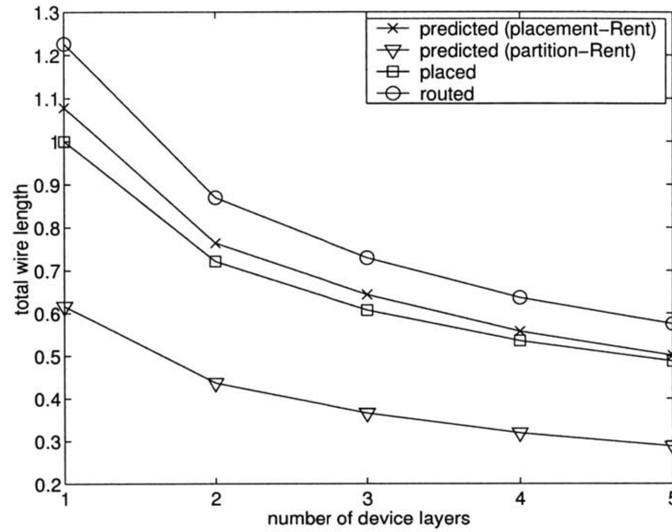


Figure 3-5: Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is given relative to the 2-D placed wire length. Inter-layer pitch t_z is 1 micrometer.

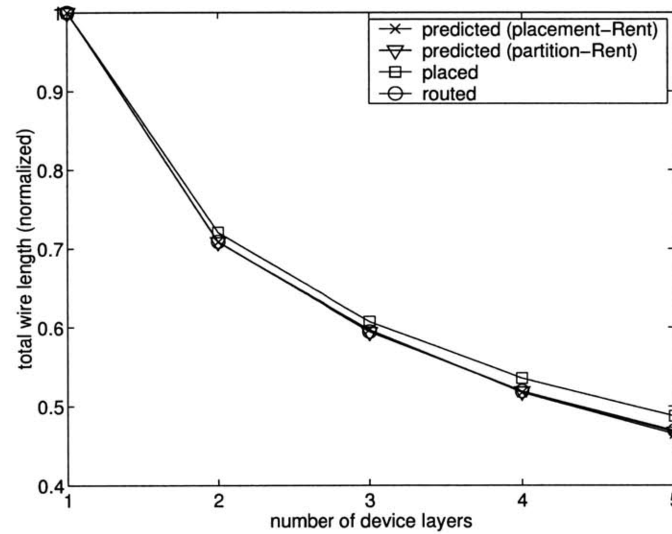


Figure 3-6: Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is normalized to exhibit the percentage reduction due to 3-D integration. Inter-layer pitch t_z is 1 micrometer.

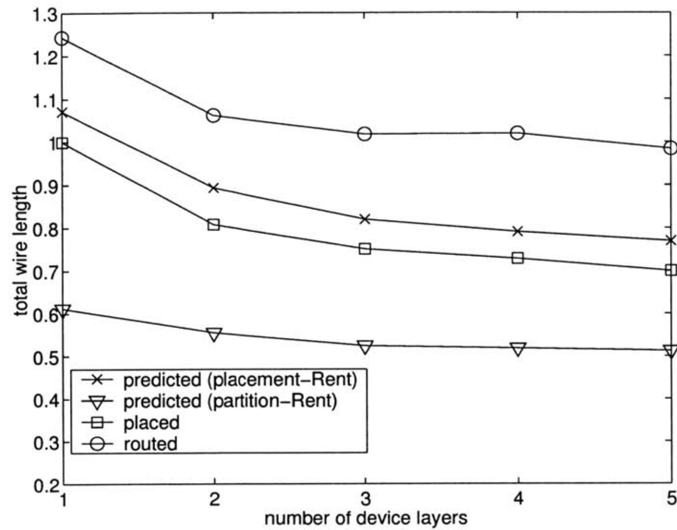


Figure 3-7: Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is given relative to the 2-D placed wire length. Inter-layer pitch t_z is 250 micrometers.

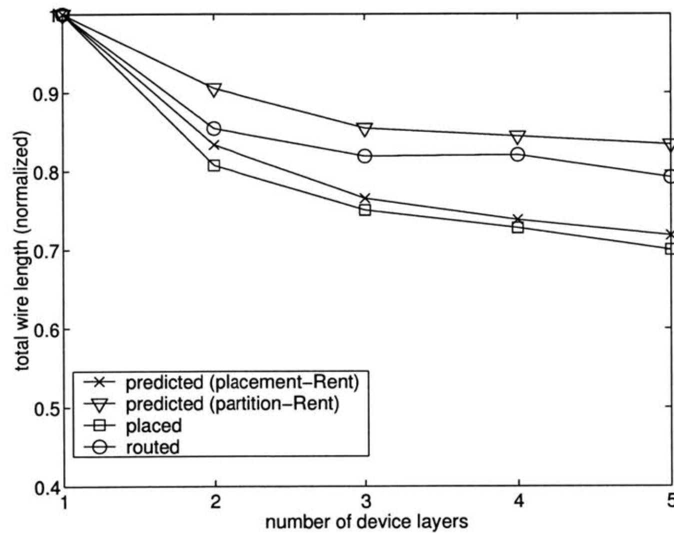


Figure 3-8: Predicted vs. placed and routed wire lengths of the average benchmark circuit. Wire length is normalized to exhibit the percentage reduction due to 3-D integration. Inter-layer pitch t_z is 250 micrometers.

	one	two	three	four	five
$t_z = 1$	21.2%	19.1%	21.3%	20.0%	18.6%
$t_z = 250$	21.2%	20.9%	20.1%	19.1%	20.1%

Table 3.3: Absolute prediction error relative to placed wire length as a function of number of device layers and inter-layer thickness.

	one	two	three	four	five
$t_z = 1$	17.4%	17.0%	17.8%	18.1%	17.9%
$t_z = 250$	18.2%	18.1%	20.1%	22.5%	21.8%

Table 3.4: Absolute prediction error relative to routed wire length as a function of number of device layers and inter-layer thickness.

reduction in total wire length of up to 28% using two device layers to 51% using five is possible.

However, there are discrepancies. Figures 3-3 and 3-4 show that the model tends to underestimate the number of medium-length wires while overestimating the number of global wires. It is believed that this discrepancy arises from the assumption within the model that the gates are laid out in a square array, whereas in actual placements, the aspect ratio may deviate from unity. Additionally, the use of a constant fan-out, independent of wire length, may affect the predicted distribution.

Another small discrepancy shows itself in the prediction of percentage reduction in wire length (Figures 3-6 and 3-8). It is not clear, however, whether the error lies in strictly two-dimensional aspects of the model, the extension to 3-D or both.

Therefore, we examine the percentage of interconnects that span multiple device layers. Figure 3-9 shows that placements with a high inter-layer pitch use less of the available inter-layer bandwidth. However, within the Rahman model, the division of interconnects into 2-D ($\beta_0 M_{2D}(l)$) and 3-D ($\sum_{i=1}^{N_z-1} \beta_i M_{2D}(l-it_z)u(l-it_z)$) components is less strongly dependent on the number of wafers. This error may possibly be explained by the computation of $M_{3D}(l)$.

We conjecture that β_i should be a function of t_z. In practice, the coefficients β_i are set discretely, based on the range of inter-layer interconnects. Specifically, for $0 < i \leq r_{max}$, $\beta_i = 2(N_z - i)$, where a third-dimension wire is permitted to span at most r_{max} device layers; for all other $i > 0$, $\beta_i = 0$. (We have assumed that $r_{max} = N_z - 1$, i.e. that an inter-layer interconnect may range over all device layers.) β_i is thus independent of t_z. In contrast, our placement tool adjusts the point of partitioning into device layers (and thus

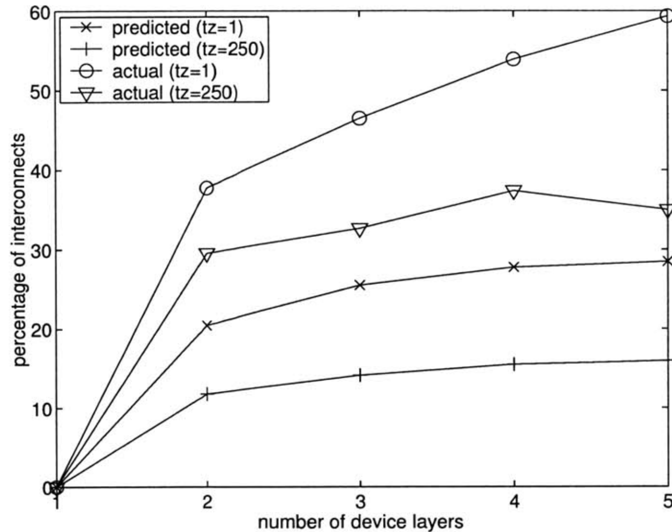


Figure 3-9: Predicted percentage of interconnects that span multiple device layers, compared with placement and routing data for $t_z = 1$ and $t_z = 250$.

the density of inter-layer interconnects) based on t_z to minimize wire length. It is not clear, however, how one may determine *a priori* the dependence of β_i on t_z.

Nevertheless, Figures 3-5 through 3-8 show that the Rahman model is a valid extension of the Davis model to 3-D integrated circuits. It proves useful for determining system-level performance characteristics of circuits that are targeted for 3-D integration.

3.3.3 Further Analyses via PR3D

Having a placement and routing tool for three-dimensional integrated circuits immediately makes two studies possible. First, we may make wire-length predictions as described in Section 3.3.2, but with inherently greater accuracy than is available with computational models. Second, we may make analyses in the areas in which the above models are less reliable.

In the first study, we analyze the placement and routing of benchmark circuits. As before, the two independent variables of interest are the number of device layers and the parasitic capacitance associated with the inter-layer interconnect. Figure 3-10 shows the total wire length, determined from placement, of the average circuit as a function of the number of device layers. Figure 3-11 shows the results of the same study done for routing. We observe that for a conventional, 2-D placement, a 27% to 51% reduction in total wire length is possible by using two to five device layers, respectively. Furthermore, the four

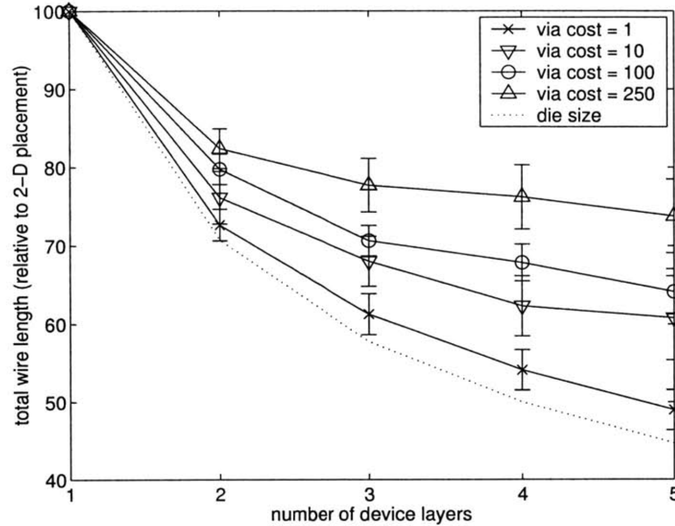


Figure 3-10: Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from placement. Total wire length is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire.

curves in each figure show quantitatively how the benefit of 3-D integration decreases as the inter-layer via capacitance is increased. This result dictates the degree to which performance tuning of the technology is required.

For the purpose of comparison with future CAD tools for 3-D ICs, we provide the full placement and routing data for this analysis in Table 3.5.

Our second analysis concerns the study in which we minimize the number of inter-layer vias. This analysis cannot be done accurately with current models, as the use of inter-layer vias by placement and routing is not yet well-understood from a theoretical standpoint (see Figure 3-9). However, using PR3D, we may ascertain what performance improvements can be obtained if we desire to avoid the use of inter-layer vias whenever possible. This situation may arise if, for example, alignment tolerances necessitate large via-to-via pitch for an otherwise low-parasitic via, or if larger interconnects such as solder bumps are used.

Figures 3-12 and 3-13 demonstrate that this approach is not substantially beneficial. Total wire length may be reduced by only 7% to 17% using two to five device layers. However, when compared with Figures 3-10 and 3-11, performance improvement in Figures 3-12 and 3-13 is more immune to inter-layer capacitance variation.

Figures 3-14 and 3-15 show the length of the longest wire from the same two analyses. We observe that this wire may be reduced by up to 31% to 56% in length. Additionally,

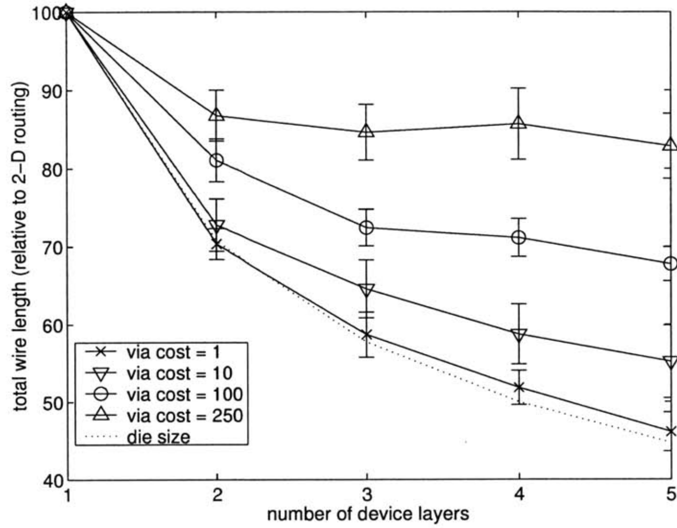


Figure 3-11: Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from routing. Total wire length is minimized by the routing tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire.

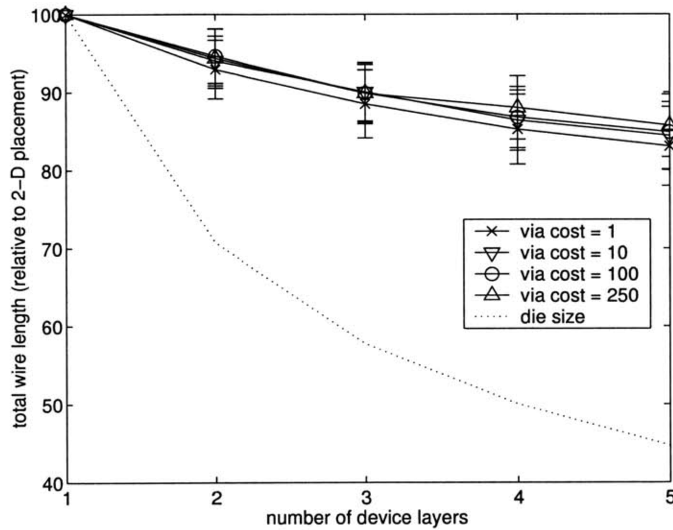


Figure 3-12: Total wire length (as a function of number of device layers) for various inter-layer via capacitances, obtained from placement. The number of inter-layer vias is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire.

		one wafer			two wafers					three wafers			
		placed	routed		placed		routed			placed		routed	
ibm01		5.70e6	7.13e6		4.12e6	28%	4.89e6	31%		3.62e6	36%	4.23e6	41%
ibm02		1.49e7	1.93e7		1.11e7	25%	1.41e7	27%		9.22e6	38%	1.13e7	41%
ibm03		1.43e7	1.78e7		1.01e7	29%	1.23e7	31%		8.77e6	39%	1.02e7	43%
ibm04		1.82e7	2.29e7		1.32e7	27%	1.56e7	32%		1.10e7	39%	1.29e7	44%
ibm05		4.00e7	5.34e7		3.11e7	22%	3.95e7	26%		2.79e7	30%	3.47e7	35%
ibm06		2.23e7	3.00e7		1.63e7	27%	2.12e7	29%		1.35e7	40%	1.68e7	44%
ibm07		3.57e7	4.48e7		2.59e7	27%	3.02e7	33%		2.10e7	41%	2.43e7	46%
ibm08		3.89e7	5.06e7		2.85e7	27%	3.58e7	29%		2.32e7	40%	2.86e7	43%
ibm09		3.15e7	3.90e7		2.25e7	29%	2.63e7	32%		1.89e7	40%	2.21e7	43%
ibm10		7.05e7	8.38e7		5.03e7	29%	5.91e7	30%		4.20e7	40%	4.91e7	41%
ibm11		4.87e7	5.89e7		3.60e7	26%	4.21e7	29%		3.05e7	37%	3.54e7	40%
ibm12		8.15e7	1.01e8		5.97e7	27%	7.22e7	29%		5.30e7	35%	6.67e7	34%
ibm13		5.93e7	7.34e7		4.28e7	28%	5.23e7	29%		3.62e7	39%	4.33e7	41%
ibm14		1.38e8	1.69e8		9.90e7	28%	1.19e8	30%		8.12e7	41%	9.70e7	43%
ibm15		1.50e8	1.85e8		1.11e8	26%	1.35e8	27%		9.03e7	40%	1.09e8	41%
ibm16		1.97e8	2.51e8		1.46e8	26%	1.80e8	28%		1.19e8	40%	1.45e8	42%
ibm17		3.00e8	3.73e8		2.09e8	30%	2.54e8	32%		1.79e8	40%	2.18e8	42%
ibm18		2.16e8	2.70e8		1.54e8	29%	1.86e8	31%		1.30e8	40%	1.53e8	43%
avg.						27%		30%			39%		41%
					four wafers					five wafers			
					placed		routed			placed		routed	
ibm01					3.16e6	45%	3.66e6	49%		2.96e6	48%	3.24e6	55%
ibm02					8.41e6	43%	1.03e7	47%		7.18e6	52%	8.72e6	55%
ibm03					7.37e6	48%	8.85e6	50%		6.93e6	52%	8.32e6	53%
ibm04					9.86e6	46%	1.18e7	48%		8.88e6	51%	1.05e7	54%
ibm05					2.50e7	38%	3.07e7	42%		2.31e7	42%	2.78e7	48%
ibm06					1.15e7	49%	1.44e7	52%		1.04e7	54%	1.24e7	59%
ibm07					1.88e7	47%	2.20e7	51%		1.68e7	53%	1.95e7	56%
ibm08					2.06e7	47%	2.51e7	50%		1.82e7	53%	2.22e7	56%
ibm09					1.68e7	47%	2.00e7	49%		1.51e7	52%	1.73e7	56%
ibm10					3.83e7	46%	4.41e7	47%		3.34e7	53%	3.85e7	54%
ibm11					2.66e7	45%	3.06e7	48%		2.50e7	49%	2.96e7	50%
ibm12					4.63e7	43%	5.56e7	45%		4.09e7	50%	4.83e7	52%
ibm13					3.22e7	46%	3.78e7	48%		2.92e7	51%	3.39e7	54%
ibm14					7.11e7	49%	8.41e7	50%		6.72e7	51%	7.99e7	53%
ibm15					8.02e7	47%	9.71e7	47%		7.22e7	52%	8.21e7	56%
ibm16					1.07e8	46%	1.33e8	47%		9.56e7	52%	1.16e8	54%
ibm17					1.57e8	48%	1.93e8	48%		1.42e8	52%	1.75e8	53%
ibm18					1.09e8	50%	1.31e8	51%		1.02e8	53%	1.21e8	55%
avg.						46%		48%			51%		54%

Table 3.5: Placement and routing data for the ISPD '98 benchmark suite. Wire lengths are in μm. Percentages are reductions relative to the one-wafer case.

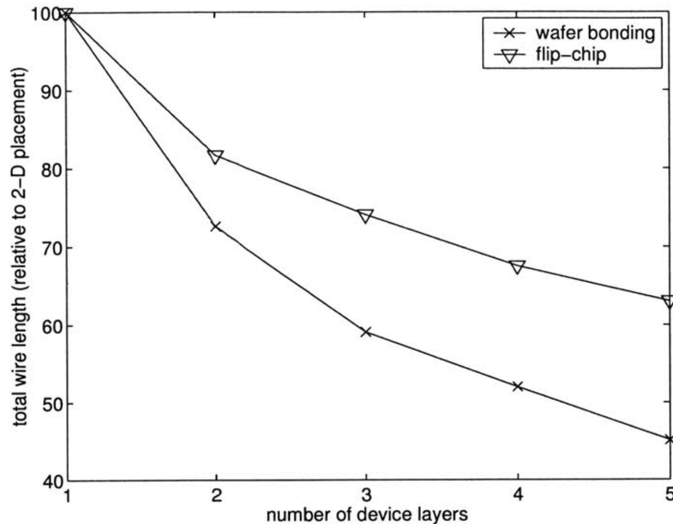


Figure 3-16: Total wire length (as a function of number of device layers) of the ibm03 benchmark circuit, using vias vs. using flip-chip solder bumps for the inter-layer interconnect.

wire-length behavior.

However, since actual circuit layout is ultimately the most reliable source of data for 3-D IC performance, we used our design tools to conduct further analyses of 3-D ICs. We found that total wire length may be improved by up to 27% to 51% using two to five device layers, but that this improvement may be attenuated if the inter-layer parasitic is increased. Similarly, we determined that the length of the longest wire in a given circuit may be reduced by up to 31% to 56% using two to five device layers, and that this performance increase is largely independent of inter-layer parasitic values.

In the following chapter, we analyze how these wire-length improvements affect more directly relevant circuit metrics such as delay and energy. In addition, we examine other important metrics such as thermal performance in 3-D ICs.

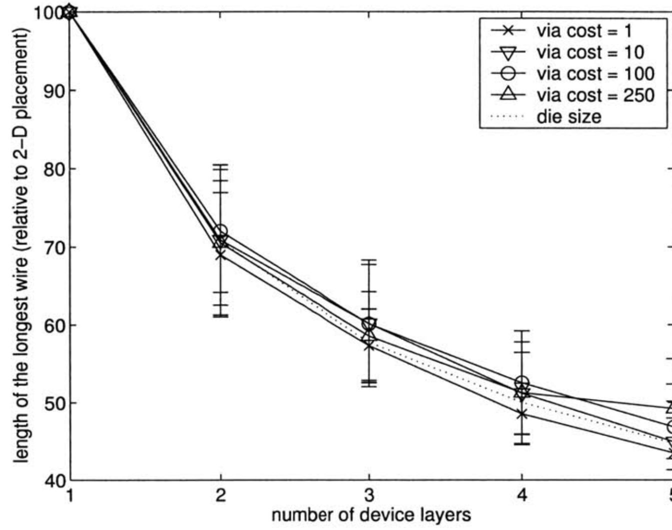


Figure 3-14: Length of the longest wire (as a function of number of device layers) for various inter-layer via capacitances. Total wire length is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire.

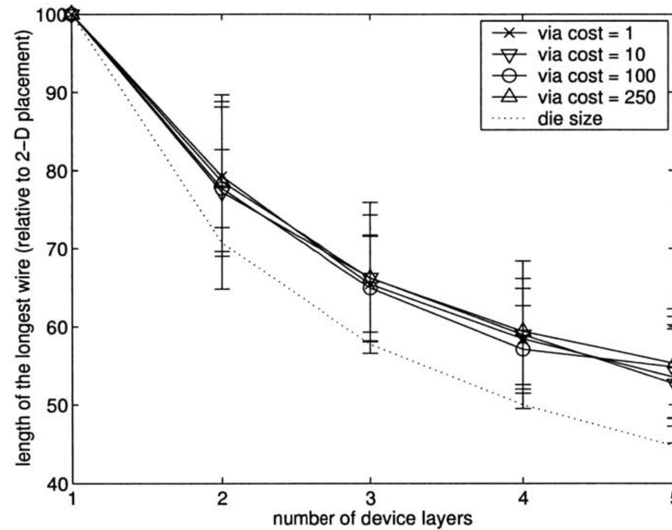


Figure 3-15: Length of the longest wire (as a function of number of device layers) for various inter-layer via capacitances. The number of inter-layer vias is minimized by the placement tool. Via cost is the via capacitance expressed relative to the capacitance of one micrometer of metal wire.

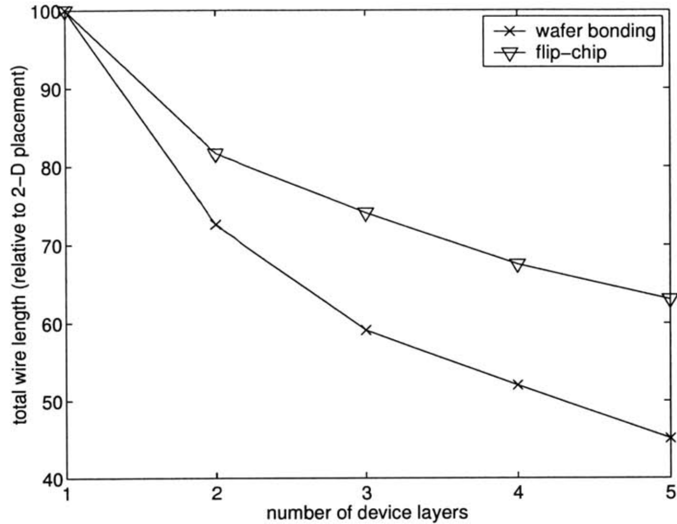


Figure 3-16: Total wire length (as a function of number of device layers) of the ibm03 benchmark circuit, using vias vs. using flip-chip solder bumps for the inter-layer interconnect.

wire-length behavior.

However, since actual circuit layout is ultimately the most reliable source of data for 3-D IC performance, we used our design tools to conduct further analyses of 3-D ICs. We found that total wire length may be improved by up to 27% to 51% using two to five device layers, but that this improvement may be attenuated if the inter-layer parasitic is increased. Similarly, we determined that the length of the longest wire in a given circuit may be reduced by up to 31% to 56% using two to five device layers, and that this performance increase is largely independent of inter-layer parasitic values.

In the following chapter, we analyze how these wire-length improvements affect more directly relevant circuit metrics such as delay and energy. In addition, we examine other important metrics such as thermal performance in 3-D ICs.

Chapter 4

Performance Characteristics of 3-D ICs

4.1 Overview

As stated in Chapter 1, interconnect performance in current and future technology generations is an increasingly dominant component of total circuit performance. Figure 1-1 shows that the delay of a medium-length wire is already beginning to exceed the delay of a typical gate. This indicates that current architectures, organized around logic gates and their devices, will not scale as required for future generations.

Similarly, Table 1.1 exhibits data for desired power consumption in future-generation microprocessors. Using ITRS data for microprocessor clock frequency, power supply voltage, total wiring per chip, and individual-wire feature sizes, and assuming an average zero-to-one transition probability of 5% for each wire, we can estimate the total interconnect power dissipation in a microprocessor at each generation. Figure 4-1 compares this estimate with the desired total power consumption. Clearly, current design techniques will not produce adequate solutions, and improvements in architecture and design methodology must be brought to bear.

We have seen that three-dimensional integration offers significant improvement in circuit wire-length metrics. For example, total wire length may be reduced by up to 27% to 51% using two to five device layers, and the length of the longest wire may be reduced similarly by up to 31% to 56%. However, timing and energy consumption are of more direct importance

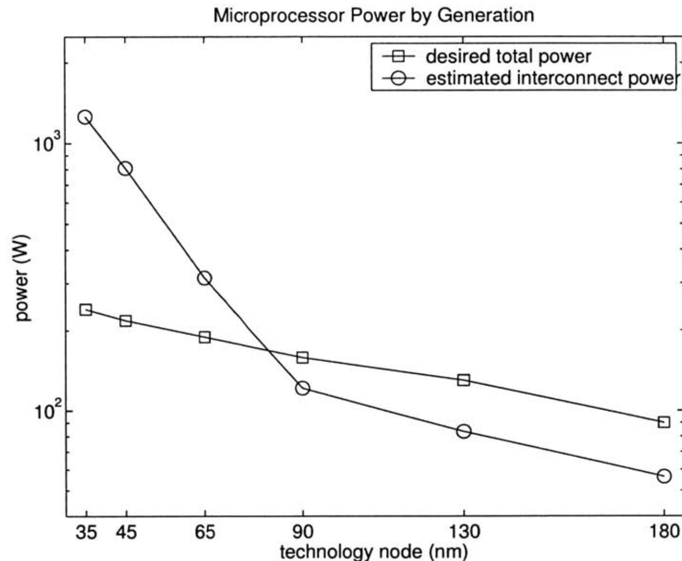


Figure 4-1: Power consumption for a high-performance microprocessor at various technology generations.

in circuit design.

The cycle time of a custom circuit may be optimized at the logic synthesis, placement, and routing stages of design. Topological optimizations can be performed during technology-independent logic synthesis [83], as well as technology mapping [84]. After mapping, the real work of timing optimization begins, as routing information becomes available to design tools [85, 86].

Similarly, the energy consumption of a custom circuit may be influenced at several stages of the design process [87, 88]. High-level architectural choices, such as block duplication and the use of sleep signals, standard-cell selection involving various drive strengths, and the use of specialized cells with multiple threshold and power-supply voltages all have been proposed to tackle various aspects of the energy consumption problem. Energy must be managed at every stage.

During placement, one of two priorities is typically seen: the best possible timing performance is desired, or some minimum timing criterion must be satisfied. Energy optimization, which may be the overall design goal, is usually performed secondarily since in most cases there is no restrictive maximum-energy constraint.[1]

One expects that 3-D integration will provide benefits in both timing-driven and energy-

[1]This is not necessarily true in future technology generations, even for timing-optimized circuits, as thermal considerations may result in global and local power constraints.

driven cases. In the timing-driven case, wires contribute to delay according to their resistance and capacitance, both functions of length. Thus, wires along the most critical timing paths must be shortened in order to enhance cycle time. In the energy-driven case, all wires contribute to energy consumption through their capacitance and the rate at which they are switched by the logic. Therefore, wires must be shortened in prioritized order according to their switching rates.

In this chapter, we show how the above percentage reductions in total wire length and length of the longest wire translate into reductions in cycle time and energy consumption.[2] We describe additional capabilities that we have added to PR3D, our placement and routing tool, for optimization and constraint of delay and energy, and we use PR3D to analyze the timing and energy performance of a set of sample circuits.

4.2 Tool Adaptations for Performance-Driven Design

We focus on the interconnect-related components of delay and energy consumption that can be affected by placement-based optimization. At current technology nodes, switched capacitance dominates the energy consumption of digital ICs. Furthermore, this capacitance comes increasingly from wires. Since 3-D integration achieves a fundamental shift in the distribution of wire lengths, an energy strategy that focuses on minimizing switched capacitance will be useful for evaluating 3-D ICs. Placement is a natural stage at which to perform this type of wire-length optimization.[3] Concurrently, timing optimization can be performed using conventional methods [85, 86, 91–93].

Thus, we have implemented four modes of operation in PR3D:

- *wire-length driven mode* – this is the conventional mode of Chapter 2;
- *timing-driven mode* – placement is optimized for least cycle time;
- *energy-driven mode* – placement is optimized for least interconnect energy consumption;

[2]Prior works on the topic of energy consumption in 3-D ICs do exist [89, 90]. However, these contain several fundamental assumptions that have proved incorrect, such as (1) a restriction in 3-D ICs to two metal layers per device layer, (2) the use of aluminum interconnect, and (3) only a minimal (25%) contribution of interconnect energy to total energy dissipation. For these reasons, we believe that the topic of 3-D IC performance characterization is an open issue.

[3]Overviews of CAD techniques for energy optimization are presented by Devadas and Malik [88] and Pedram [87].

- *timing-constrained mode* – some metric (such as energy consumption) is optimized under a supplied timing constraint.

As Chapter 2 describes, the core placement algorithm is refinement by recursive bisection of the net list. Specifically, the circuit net list is represented by a hypergraph, with standard cells becoming nodes and wires becoming hyperedges. The die area is partitioned recursively into halves such that the number of nets crossing any partition is minimized [48].

To optimize energy performance, we extend the placement algorithm to include switching activity. Specifically, the energy consumption of a net i is given by

$$E_i = N_i \left(C_{is} + \sum_{j \neq i} M_{ij} C_{ij} \right) V_{DD}^2, \tag{4.1}$$

where N_i is the number of 0-to-1 transitions, C_{is} is the capacitance of the net to the substrate, C_{ij} is the coupling capacitance to net j, M_{ij} is a Miller factor that accounts for signal correlations between nets i and j, and V_{DD} is the supply voltage. The switching activity is given by the average number of transitions per unit time or per cycle.

Since the capacitance C_{is} essentially follows the net length, the energy consumption may be reduced by weighting each net according to its activity. We augment our placement tool to minimize the weighted sum of the nets crossing a partition. Thus, nets with high activity are less likely to be cut by a partition. This leads to high-activity nets being very localized and therefore shorter and less capacitive. (The coupling capacitance C_{ij}, while important in computing energy consumption, is difficult to determine before routing is complete. However, it is generally valid to assume that reducing the lengths of highly-active wires will not lead to an increase in coupling-capacitance energy dissipation.)

At the same time, we also extend PR3D to manage timing performance during placement. We utilize a combination of net-based and path-based approaches [91,92]. Separate approaches are employed for timing optimization and for timing constraint.

For timing optimization, we use a standard path-based counting technique: We seek to minimize both net cut and path cut during recursive bisection. Nets are weighted according to the number of critical paths on which they lie. If a given path exceeds a fixed number of path cuts, the nets on that path are prohibited from being cut further.[4]

[4] There exist further placement-based optimizations that may enhance interconnect-dominated circuit timing characteristics. For example, repeaters may be inserted in long wires in order to reduce wire delay.

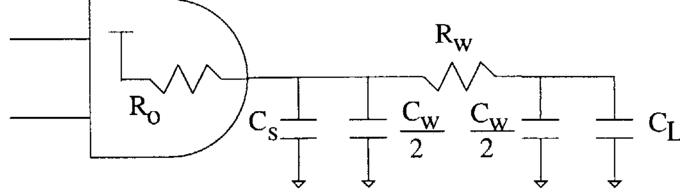


Figure 4-2: Delay model for gates and wires.

Conversely, for timing-constrained optimization, delay is not a component of the cost function. We therefore seek to minimize net cut weighted as in wire-length or energy-optimized modes (i.e. unweighted net-cut or net-cut weighted by switching activity). However, we insert a timing-analysis step between partitionings; if any critical path exceeds 95% of its allotted delay, the nets on that path are prohibited from further cuts.

For delay calculation, we use an Elmore delay model as depicted in Figure 4-2 for a two-point net. R_0 is the cell output resistance and R_w is the wire resistance; C_s is the cell output capacitance, C_w is the wire capacitance, and C_L is the load capacitance. Under this model, the total delay is

$$\tau_d = R_0 \left(C_s + C_w + C_L\right) + R_w \left(\frac{C_w}{2} + C_L\right). \tag{4.2}$$

The R_0 component and output and load capacitances are determined using table data from the cell vendor [37]. The wire resistance and capacitance are calculated with a scaled half-perimeter metric. Specifically, the 3-D bounding box is utilized as in Chapter 2 to estimate the total wire length of a net, but here the lateral dimensions are scaled by a resistance or capacitance per-unit-length factor and the third dimension is scaled by the inter-wafer interconnect resistance or capacitance.

4.3 Methodology and Circuits Under Test

To evaluate the effectiveness of our optimization methodologies, we placed and routed three circuits. For each circuit, we obtained four layouts that correspond to the four operational modes described in Section 4.2. The circuits were supplied in Verilog format, which we

Furthermore, the optimal number of repeaters for a given circuit decreases substantially if more than one wafer is used [94]. However, at the current technology node, repeater insertion is necessary only for the largest chips, and is not required for the circuits we study here. We will take into account repeater insertion in Chapter 6, where we analyze 3-D integration in future technology nodes.

	number of cells	number of nets	layout area (2-D case)
FFT	7181	7969	442.86 μm × 441.84 μm
DES	19673	20563	722.70 μm × 721.84 μm
MAC	26844	27246	978.78 μm × 978.32 μm

Table 4.1: Relevant parameters for the circuits in this study.

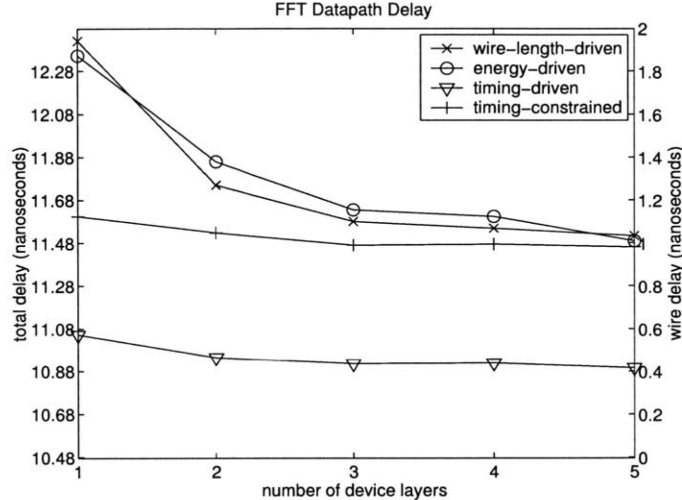


Figure 4-3: Cycle time of an FFT datapath using various placement modes.

compiled to cells with Synopsys Design Compiler. During this synthesis, we supplied Design Compiler with the timing constraint that we subsequently used for energy optimization by PR3D. We also assessed the activity factors of the nets in the design with Design Compiler by using a number of representative test inputs in gate-level simulation. The activity factors were produced in SAIF format and imported into PR3D. Once layout was generated, extraction was performed on the layout, and the resulting transistor-level net list was simulated using Synopsys NanoSim.

The three circuits tested are a Fast-Fourier-Transform (FFT) datapath circuit provided by Alice Wang of MIT [95], a Data Encryption Standard (DES) cryptographic core obtained from `opencores.org` [96], and a 64-bit multiplier-accumulator (MAC) from the ISPD '01 benchmark suite [93]. Table 4.1 provides relevant data for the circuits.

4.4 Timing Characteristics of 3-D ICs

Figures 4-3 through 4-5 exhibit the nature of cycle-time improvement with 3-D integration. The impact of additional device layers on cycle time is dependent on the optimization

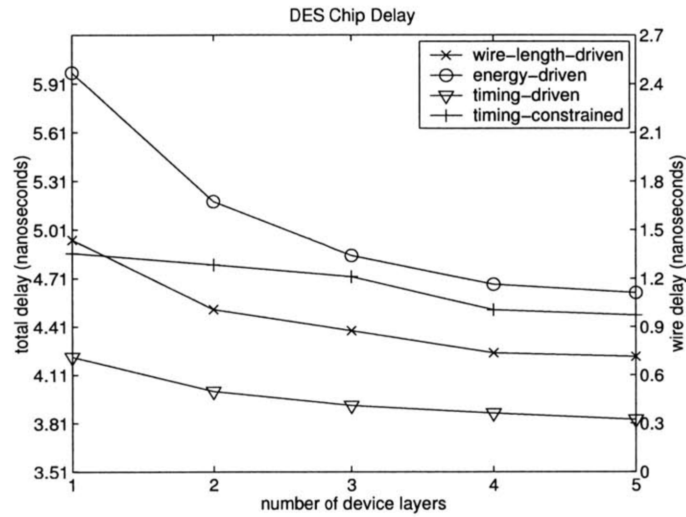


Figure 4-4: Cycle time of a DES implementation using various placement modes.

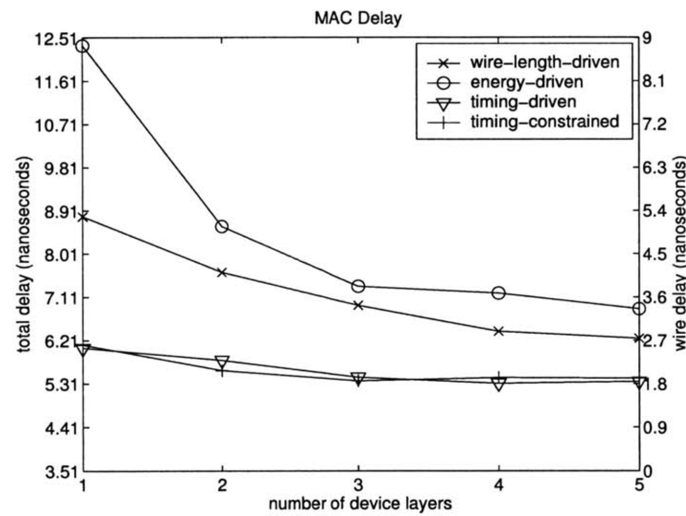


Figure 4-5: Cycle time of a 64-bit MAC using various placement modes.

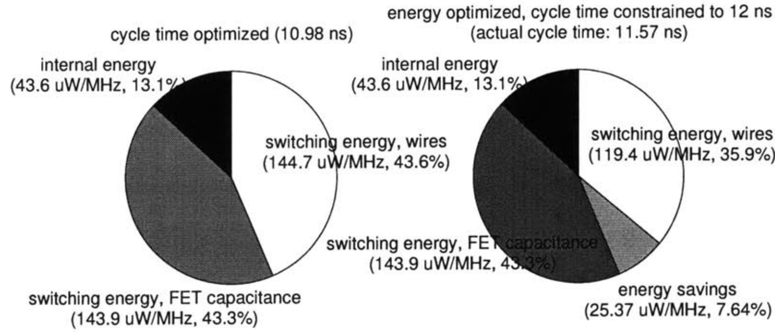


Figure 4-6: Energy consumption of an FFT datapath in timing-optimized vs. timing-constrained placement.

mode. For example, while 50% of interconnect delay can be eliminated in energy-driven cases using up to five wafers, with as much as a factor-of-three reduction in the MAC circuit, the improvement can be as little as 30% in the timing-optimized case. This is due to the fact that in a timing-optimized 2-D circuit, long wires are relegated to non-critical paths; therefore, the impact of 3-D wire-length reduction on critical paths is less profound.

However, we can make two general observations. First, the improvement of secondary metrics by 3-D integration is not to be ignored. We will show in Section 4.5 that similar results are achieved for energy consumption in timing-optimized designs. This leads us to conclude that in overall figure-of-merit measurements such as energy-delay product, 3-D integration will yield significant benefits. Second, we observe that in the larger circuits, the impact of 3-D integration on cycle time, even in timing-optimized designs, is greater. It is likely that the larger interconnect structures in these circuits result in greater overall performance improvements in 3-D.

4.5 Energy Characteristics of 3-D ICs

4.5.1 Energy Performance of the Conventional Circuits Under Test

To understand the impact of 3-D integration on circuit energy consumption, we must consider the role of interconnect energy as a part of total energy dissipation. We must also take into account the ability of timing-constrained energy optimization to improve the interconnect energy consumption in conventional (2-D) circuits.

Figure 4-6 shows the energy consumption of the 32-bit Fast-Fourier-Transform (FFT) datapath. In the graph on the left, switched-capacitance energy dissipation accounts for

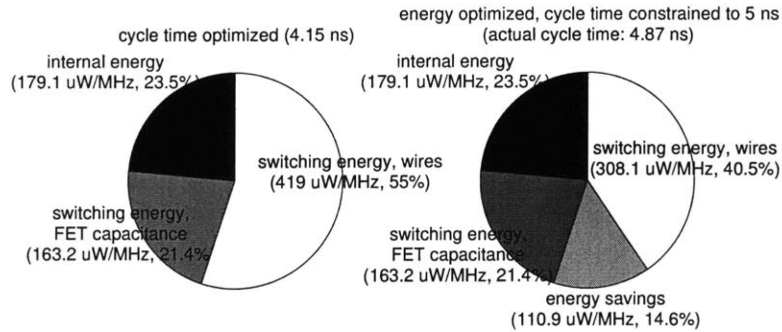


Figure 4-7: Energy consumption of a DES chip in timing-optimized vs. timing-constrained placement.

approximately 87% of total energy consumption, and cell internal energy makes up the remainder. The switching energy consists of two parts. An estimated 43% is due to switching at the cell inputs and outputs (i.e. gate and source/drain capacitances), and the remaining 44% is due to wires. This layout is optimized for cycle time.

The graph on the right shows the same circuit, but here the cycle time is constrained to 12 ns, and energy is optimized by the placement tool. Although the cycle time is approximately 0.6 ns slower, it still meets the constraint. Furthermore, the wire component of energy dissipation is reduced by 18%, thereby leading to an overall reduction of 8%.[5]

Figure 4-7 shows the energy consumption of the second circuit, an implementation of the cryptographic Data Encryption Standard (DES). For this circuit, 76% of the total energy dissipation of the timing-optimized layout (as seen in the graph on the left) is due to switched capacitance. This 76% consists of 21% cell I/O switching energy and 55% wire switching energy. The graph on the right shows that while the cycle time has increased by approximately 0.7 ns (while still meeting the constraint), the interconnect energy dissipation has been reduced by 26%, thereby leading to an overall reduction in energy consumption by 15%.

Figure 4-8 exhibits the energy consumption of the third circuit, a multiplier-accumulator (MAC). For this circuit, 88% of the total energy dissipation of the timing-optimized layout is due to switching activity, where 21% represents cell I/O switching energy and 67% represents

[5]We have stated that the switching energy dissipation due to I/O FET capacitance is the same in both graphs. The astute reader will observe that signal glitching should be different in the two circuits due to differing path delays. As a result, the FET-capacitance switching energy and cell-internal energy should also differ by a small amount. However, within the simulation framework, it is difficult to isolate the cell input and output energy components from other components within the cell. Thus, we estimate the FET-capacitance switching energy using glitch-free activity factors from behavioral simulation.

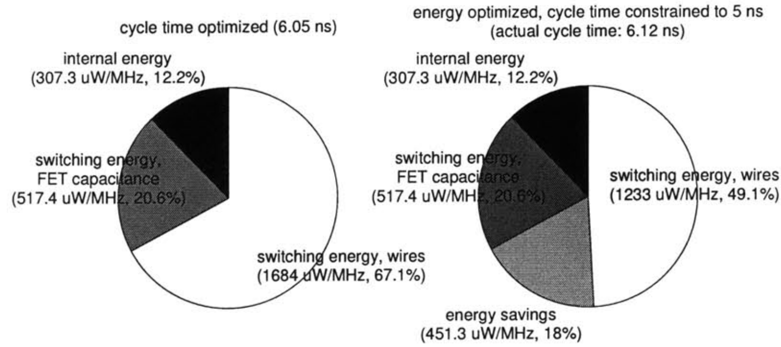


Figure 4-8: Energy consumption of a multiplier-accumulator chip in timing-optimized vs. timing-constrained placement.

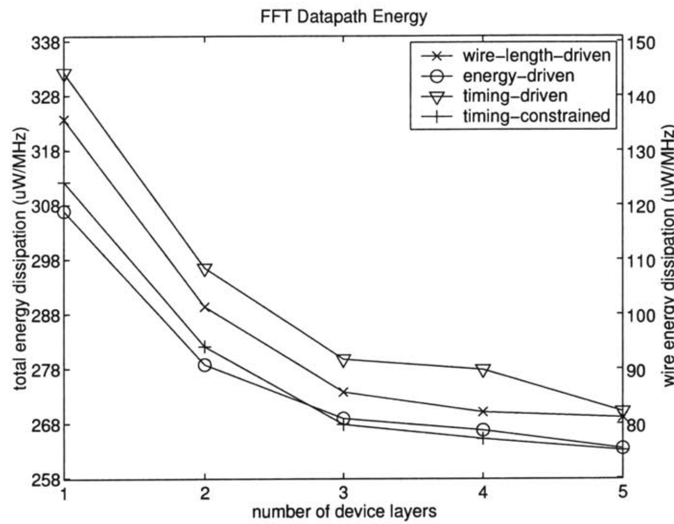


Figure 4-9: Energy consumption of the FFT datapath vs. number of wafers used for placement.

wires. In contrast, in timing-constrained, energy-optimized mode, 18% of total energy is saved with a minor amount of loss in cycle time. This savings represents 27% of the interconnect energy dissipation of the timing-optimized case.

Circuit energy dissipation therefore consists largely of interconnect switching energy. Furthermore, we can trade off cycle-time optimization for energy optimization even in 2-D circuits. We now examine how these trade-offs scale as we add additional wafers.

4.5.2 Energy Optimization in 3-D

Figure 4-9 shows the manner in which the interconnect energy dissipation of the FFT datapath circuit scales with the number of wafers. Four cases that correspond to the four operational modes described in Section 4.2 are shown. In the timing-constrained, energy-

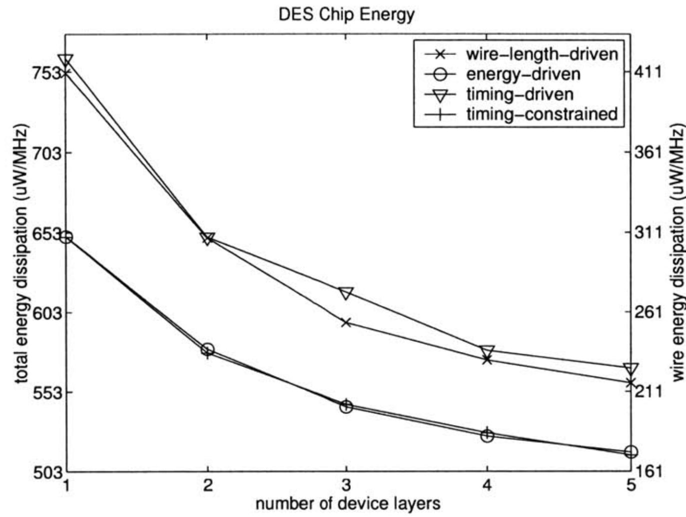


Figure 4-10: Energy consumption of the DES chip vs. number of wafers used for placement.

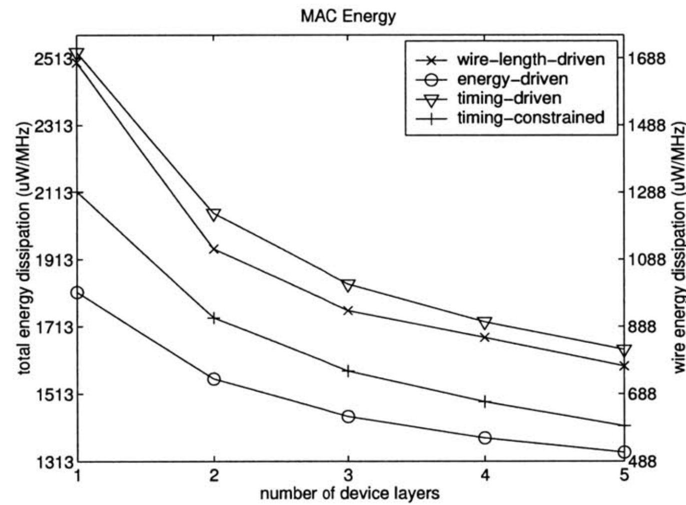


Figure 4-11: Energy consumption of the 64-bit MAC vs. number of wafers used for placement.

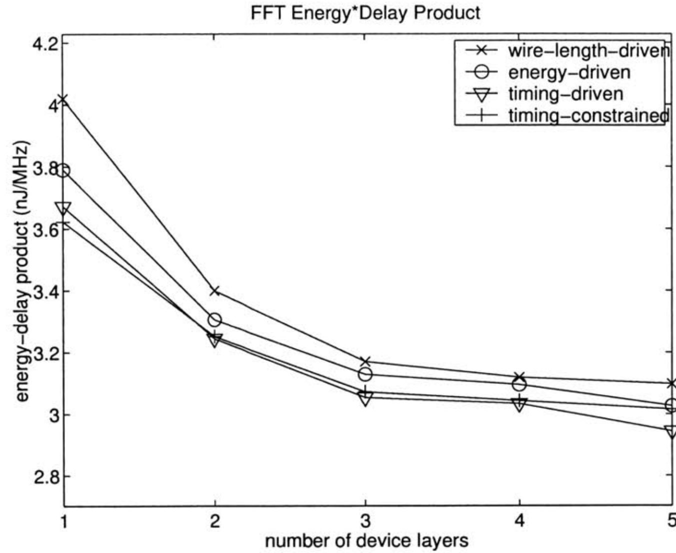


Figure 4-12: Energy-delay product for the FFT datapath vs. number of wafers used for placement.

driven mode, the FFT datapath cycle time is constrained to 12 ns. We observe that in this mode, we are able to reduce interconnect energy consumption 24% to 39% using two to five wafers respectively. This is in addition to the savings that can be realized relative to timing-driven mode. We can reduce the interconnect energy consumption of a timing-driven 2-D design by 48% by performing timing-constrained energy optimization and using five wafers.

Similarly, Figure 4-10 shows that for the DES chip, 24% to 45% of the interconnect energy consumption of a 2-D layout can be eliminated by targeting two to five wafers respectively. In comparison to a single-wafer timing-optimized design, we can reduce interconnect energy consumption by 60% by employing timing-constrained energy optimization and using five wafers.

Figure 4-11 shows that for the 64-bit MAC, 29% to 54% of the interconnect energy can be saved by using two to five wafers respectively. 65% of the interconnect energy dissipation of the single-wafer timing-optimized MAC can be eliminated with the use of five wafers and timing-constrained energy optimization.

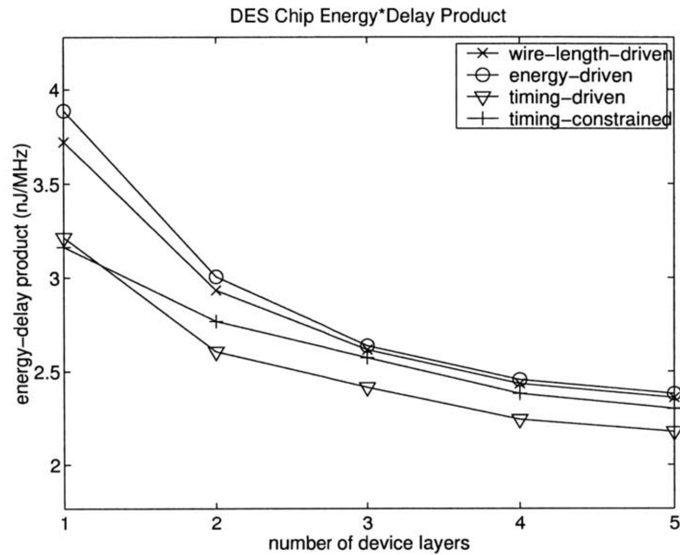


Figure 4-13: Energy-delay product for the DES chip vs. number of wafers used for placement.

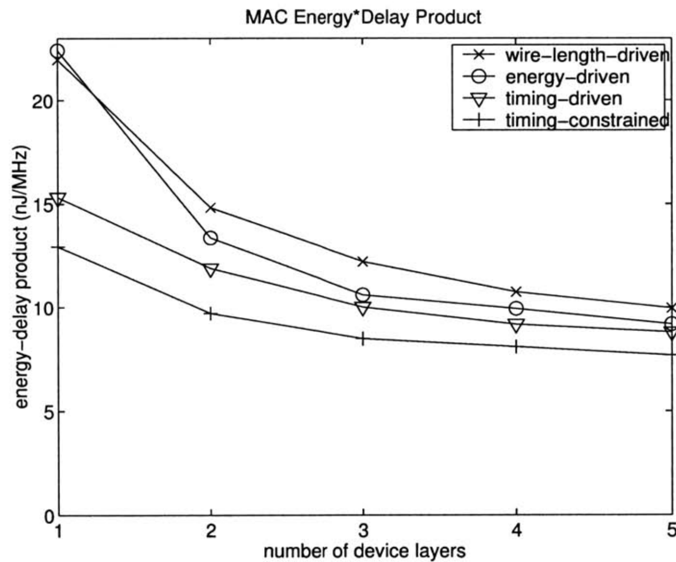


Figure 4-14: Energy-delay product for the 64-bit MAC vs. number of wafers used for placement.

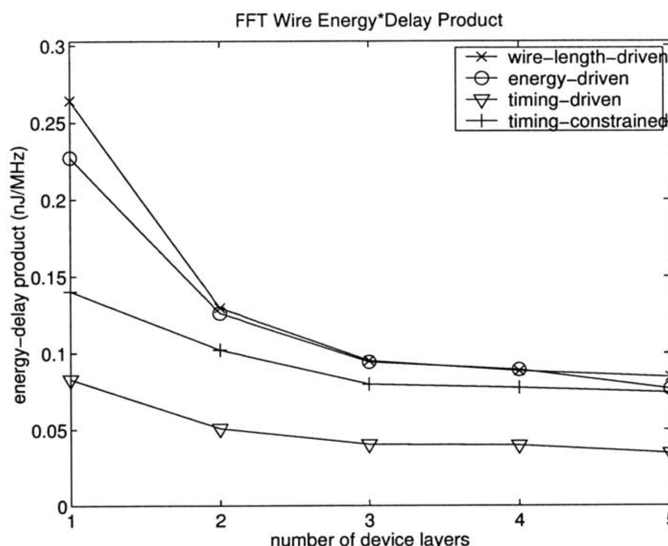


Figure 4-15: Wire energy-delay product for the FFT datapath vs. number of wafers used for placement.

4.6 Energy-Delay Product

Figures 4-12 through 4-14 show the energy-delay product for the three circuits. This product can be reduced by 20% for the FFT, 32% for the DES chip, and 41% for the MAC, using five wafers. In view of the fact that we are considering *total* energy and *total* delay, not simply the components associated with wires, this result is quite striking.

Figures 4-15 through 4-17 show the component of energy-delay product that is associated with interconnect. The impact of 3-D integration is clearly quite substantial. For the FFT, up to 58% of wire energy-delay product can be eliminated by using five wafers for integration. Similarly, 75% and 66% of wire energy-delay product can be eliminated for the DES chip and MAC, respectively. This result demonstrates that 3-D integration can have a tremendous impact on circuit performance.

4.7 Summary

The wire-length results of the previous chapter do indeed translate into similar results for circuit performance. In this chapter, we analyzed the behavior of three designs: (1) a 32-bit Fast-Fourier-Transform (FFT) datapath, (2) an implementation of the DES cryptographic algorithm, and (3) a 64-bit multiplier-accumulator (MAC) chip. We found that the impact of 3-D integration increases with larger chip sizes, and that this impact amounts to up to

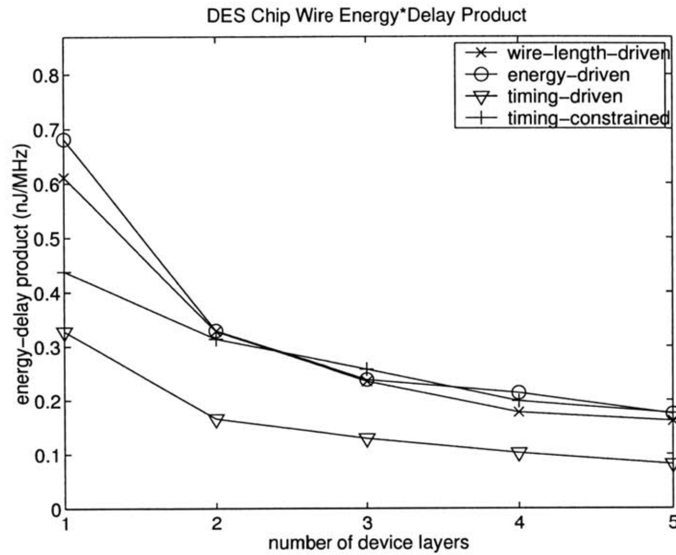


Figure 4-16: Wire energy-delay product for the DES chip vs. number of wafers used for placement.

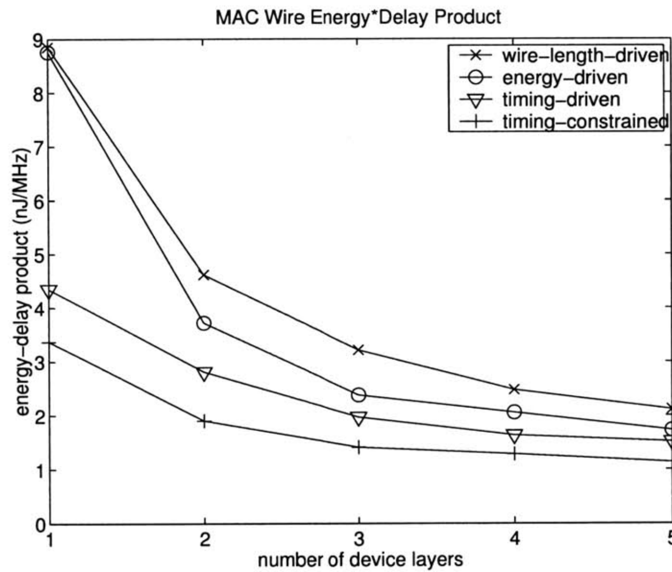


Figure 4-17: Wire energy-delay product for the 64-bit MAC chip vs. number of wafers used for placement.

a 54% reduction in wire delay, 54% reduction in interconnect energy dissipation, and 75% reduction in wire energy-delay product, all using up to five wafers.

Naturally, no improvement comes without cost. We have been concerned with performance trade-offs that might ensue from our optimization of delay and energy. While the energy profiles of these 3-D ICs may improve drastically, one potential issue is that they do not scale as well as the die footprint. Heat removal is therefore critical. The next chapter considers precisely this problem.

Chapter 5

3-D IC Thermal Management and Optimization

5.1 Motivation

It is anticipated that power requirements for high-performance microprocessors will increase exponentially with each foreseeable technology generation [5]. Conversely, it is desired to maintain zero or modest growth in the maximum die size over the same period. Therefore, without innovations in the design of circuits, integration materials, and/or packaging components, die temperatures will escalate quickly beyond any acceptable limit.

Using the ITRS-projected power requirements [5], we can determine the extent to which conventional packaging technology must be improved for future device technology generations. Figure 5-1 shows the required heat sink thermal resistance for a state-of-the-art microprocessor at each generation. Given that in late 2003, the best mass-market heat-sink technology available achieved a thermal resistance of 0.7 cm^2K/W [97], it is clear that industry is narrowly outpacing design requirements.

In addition to concerns regarding aggregate thermal behavior, circuit-level issues exist that must be considered. For example, as die temperatures increase, device performance necessarily suffers. The absolute threshold voltage of both NMOS and PMOS devices decreases by 1.0 mV/K for submicrometer devices [98]. This results in an increase in leakage power dissipation; specifically, for a chip operating at 100°C, the leakage power is ten times higher than the corresponding dissipation at room temperature [99]. Additionally,

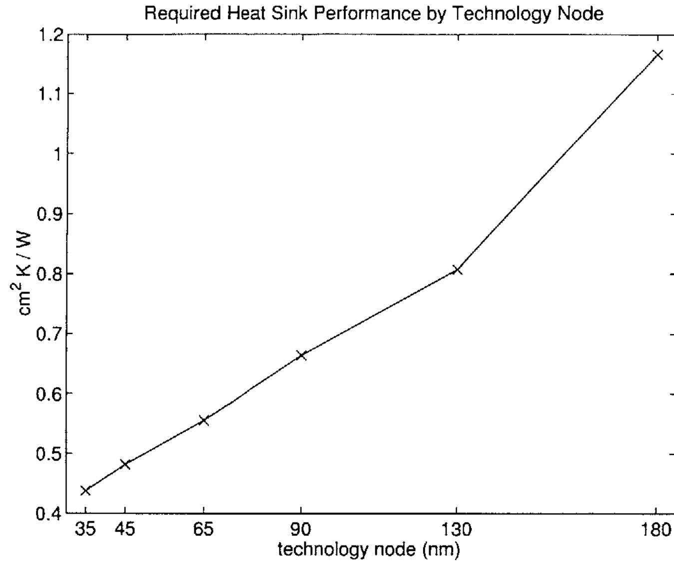


Figure 5-1: Minimum required heat sink thermal resistance by technology generation, based on ITRS projections for microprocessor size and power dissipation. The desired maximum die temperature is 100°C.

both electron and hole mobilities are reduced by increasing temperature. This dependence is of order $T^{-3/2}$ [100]; despite the decrease in threshold voltage, devices are actually slower at higher temperatures. Transistor g_m also decreases, thereby reducing gain. Moreover, reliability issues associated with die heating are also present. For example, time-dependent dielectric breakdown (TDDB) time-to-failure is exponentially dependent on $1/T$, such that at 100°C, the lifetime is reduced by four to five orders of magnitude over room temperature operation [101].[1]

Increases in die temperature also affect the performance of interconnect in a number of ways [102, 103]. First, resistance (and to a far lesser extent, capacitance) varies with temperature; interconnect delay therefore increases with temperature. Second, reliability is impacted by die heating: both interconnect lifetime, which is exponentially dependent on $1/T$ [104], and immunity to spontaneous thermally-induced open-circuit metal failure [105] require limits on the extent to which transistors and interconnects may generate heat. Third, interconnect self-heating in and of itself constrains the maximum allowable RMS current density through any given wire.

Thermal gradients, or variations in temperature along the surface of a chip, can also

[1]TDDB with or without die heating is not significant at current technology nodes. However, as gate-oxide thicknesses are scaled down in future nodes, die heating can result in a reduction of lifetime from years to mere seconds.

impact system performance. For example, under a 60°C gradient over a 1000 μm clock-tree line, clock skew can be degraded by over 5% of clock-driver-to-load delay [106], which is an amount comparable with the nominal skew [107]. Since processors such as the Alpha 21064 have exhibited 30°C differentials due to high-power clock drivers [108], the problem of thermal-gradient-induced skew is quite serious in practice.

For 3-D ICs, the complications brought on by temperature are expected to be even worse, to the extent that temperature is often considered a major hindrance for 3-D integration [31, 32]. However, apart from first-order models, little has been known about the precise role of temperature in 3-D ICs. Prior work in this area has focused on obtaining estimates of total power consumption from which average temperature estimates have been computed [35]. As we have obtained actual power data in Chapter 4, we may make a more accurate determination. Furthermore, through the use of our placement tool to control thermal interactions at a local level, we examine the extent to which it is possible to mitigate the adverse effects of 3-D integration, with respect to both global and local die temperatures. We also analyze the use of advanced cooling technologies that interact with the electrical substrate at the micrometer scale.

In the following sections, we illustrate our advances in all three of these areas. We review a first-order model for die temperature in 3-D ICs. We then describe modifications to our CAD tool PR3D for use in optimizing the thermal profile of a 3-D placement. Employing this augmented PR3D, we analyze local and global die temperatures for a Fast-Fourier Transform (FFT) circuit under a variety of 2-D and 3-D placement conditions with and without thermal optimization. Finally, we consider the impact of a candidate advanced cooling technology on 3-D ICs; we will show that this technology can entirely alleviate the negative thermal effects introduced by 3-D integration.

5.2 First-Order Model for Die Temperature in 3-D ICs

Assuming a 3-D stack of n device layers in which the bottom-most layer is connected to a conventional package and heat sink, a first-order analytical model shows that the layer-to-layer temperature rise is proportional to the power dissipation per unit area of the chip [31, 32]. Let T_i be the average temperature of the ith device layer (T_0 is ambient) and P_i be the total power dissipation of the ith layer. We assume a layout area of A_1 for a

two-dimensional placement and that with n device layers, the area of each layer is A_1/n. The one-dimensional heat diffusion equation states that

$$T_i - T_{i-1} = R_i \sum_{k=i}^{n} \frac{P_k}{A_1/n}, \quad (5.1)$$

where R_i is the effective thermal transfer resistance from device layer i to layer $i-1$, and $R_1 = R_{hs}$ is the heat sink thermal resistance. Thus, the temperature of the uppermost die is

$$T_n = T_0 + \sum_{k=1}^{n} \left(\frac{P_i}{A_1/n} \sum_{m=1}^{k} R_m \right). \quad (5.2)$$

Considering that in most conventional packages, $R_{hs} \gg R_{2,\ldots,n}$, we may further simplify:

$$T_n = T_0 + R_{hs} \sum_{k=1}^{n} \frac{P_i}{A_1/n} = T_0 + nR_{hs}\frac{P_{tot}}{A_1}, \quad (5.3)$$

where P_{tot} is the total power dissipation of the chip [31].

From this, we can see that in 3-D ICs, the top-layer die temperature above ambient will rise linearly with the number of device layers, when absent any reduction in power dissipation due to 3-D integration. Combining Equation 5.3 with the power-reduction results from Chapter 4 allows us to predict thermal performance for 3-D ICs. For example, we postulate a 2-D circuit that dissipates 50 W over a die area of 2 sq. cm. in an ambient temperature of 25°C. If the interconnect power consumption of this chip (which we assume to be 65% of the total 2-D power) decreases in a 3-D implementation as observed in Chapter 4, and if the die area scales inversely with the number of device layers, then the temperature of the top-most device layer increases as shown in Figure 5-2.

Thermal management of 3-D ICs is therefore critical. However, in this first-order analysis, several assumptions are made that provide directions in which to alleviate thermal problems. For example, it is assumed that power dissipation is uniform among all device layers. However, cell placement can be driven such that more active cells and wires are constrained to lower device layers. Techniques for the constraint of power distribution for 2-D placement (to minimize within-die temperature variation) exist [109–111], as well as prior work on thermal-driven placement for 3-D ICs in particular [112]. However, the prior work does not consider the effect of 3-D integration on the thermal characteristics of the placement (its circuit data is confined to four-device-layer placements only), nor does it use

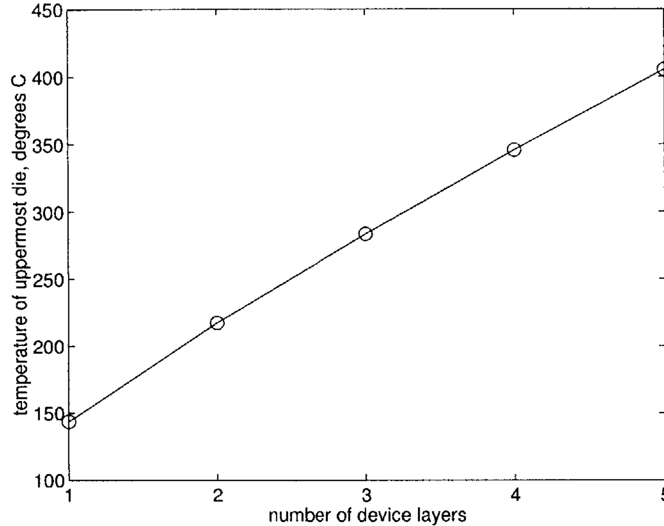


Figure 5-2: Temperature of the uppermost die in a 3-D stack, assuming 50 W power dissipation, 2 sq. cm. total circuit area, and 25°C ambient temperature.

measurement-calibrated energy data for the individual cells and wires in the placement. We examine the role of multiple-wafer integration on circuit thermal characteristics using an accurate energy model for the circuit components and packaging.

Additionally, the development of novel packaging technologies such as micro-channels for fluidic cooling [113] has been proposed for 3-D integration. We evaluate the improvement in thermal profile that can be obtained with such technologies by using our thermal-driven tools.

5.3 Placement-Based Optimization of Thermal Characteristics

We extend the methodology of Tsai and Kang [109] to optimize 3-D IC placements. Specifically, energy consumption at any given physical location in a circuit translates into a rise in temperature at that location as the energy is dissipated into the substrate as heat. The temperature distribution within any material component of a chip may be computed by the steady-state heat diffusion equation

$$k \cdot \nabla^2 T + g(x, y, z) = 0, \tag{5.4}$$

where T is the temperature distribution, g is the power density distribution, and k is the thermal conductivity of the material. This equation may be solved by the finite-difference method and discretizing the 3-D IC into an m-by-m-by-p grid of $n = m^2 p$ nodes. (We take $m = 50$ for lateral temperature resolution and p equal to the total number of distinct material layers over all wafers; extra layers are allocated for bulk materials such as the bottom substrate.) The result is a matrix equation

$$GT = P, \tag{5.5}$$

where G is an n-by-n matrix of thermal conductances connecting adjacent nodes, T is the temperature at each node, and P is the power dissipation at each node.

Given a circuit layout and operating frequency, the power dissipation P_k is known, and the temperature $T_k = G \backslash P_k$ may be computed (by the preconditioned conjugate gradient method, for example). More importantly, given a desired thermal distribution T_d, a power constraint $P_d = GT_d$ may be computed. Placement optimization of 2-D ICs using this power constraint is carried out by Tsai and Kang [109].

For 3-D ICs, we assume a conventional package in which the bottom substrate is attached to a heat spreader and heat sink. Numbering the wafers consecutively from 1 to n with wafer 1 adjacent to the sink, the average temperature of wafer i must exceed that of wafer $i-1$, because the heat from the ith wafer must flow through wafers $i-1$ through 1 before being dissipated into the sink. Therefore, if a uniform thermal distribution T_d is desired, the resulting power constraint is zero for wafers 2 through n. Thus, rather than attempt to obtain a uniform thermal distribution for the entire circuit, we focus on the within-wafer variation for each wafer. To manage wafer-to-wafer thermal gradients, we strive to place most of the energy dissipation close to the heat sink. Specifically, when partitioning a subcircuit placement into wafers i and $i+1$, energy consumption on wafer $i+1$ is minimized subject to the constraint that equal areas of standard cells are placed on each wafer.

5.4 Thermal Characteristics of 3-D ICs

Figures 5-3 and 5-4 illustrate the mechanism of our thermal optimization. Figure 5-3 shows the temperature of the uppermost die of a three-wafer FFT placement. In the energy-optimized case, a hot spot results from the shortening of the highly-active wires.

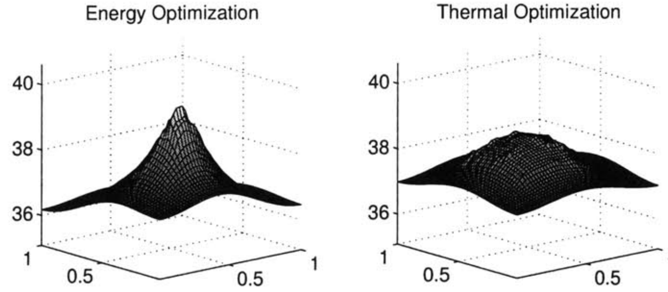


Figure 5-3: Celsius die temperature of the top wafer of a three-wafer placement of the FFT datapath.

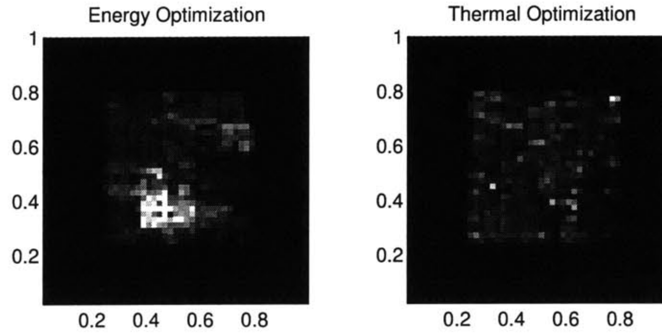


Figure 5-4: Energy distribution of the top wafer of a three-wafer placement of the FFT datapath.

As Figure 5-4 demonstrates, the origin of the hot spot is clear from the energy distribution. Thermal optimization spreads the energy consumption over the entire die; the hot spot is thereby reduced or eliminated. We assume a conventional package with a heat sink extraction capability of $R_{hs} = 1\text{cm}^2\text{K/W}$, which is achievable with currently-available technology [97]. In all analyses, the circuit is run at 80 MHz in an ambient temperature of 25°C.

Figures 5-5 through 5-10 show the thermal performance of the FFT datapath when using one to five wafers. In the first set of figures, we assume that the overall footprint of the die is unchanged as we scale the number of wafers (as may be the case in an I/O-limited situation). Figure 5-5 shows the temperature of each die for both placements. In Figure 5-6, we plot the absolute temperature difference (maximum temperature minus minimum temperature over the entire circuit) against the number of wafers used. Figure 5-7 shows the wafer-to-wafer average temperature differential (i.e. average temperature of the hottest wafer minus average temperature of the coolest wafer). Figures 5-8 through 5-10 provide the temperature differential when the overall footprint of the die scales inversely with the

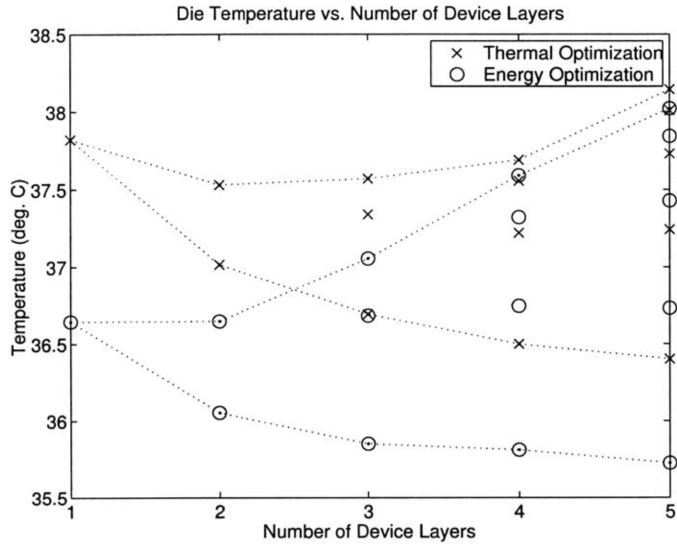


Figure 5-5: Die temperature of the FFT datapath vs. number of wafers (fixed-die case).

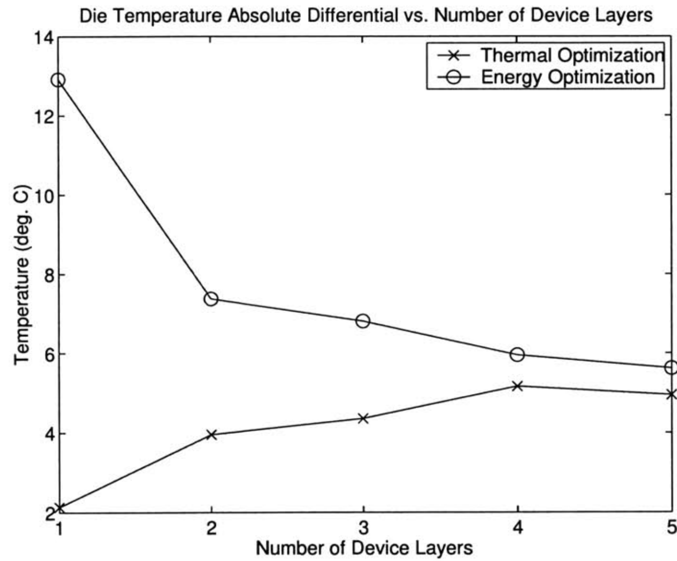


Figure 5-6: Absolute temperature differential of the FFT datapath vs. number of wafers (fixed-die case).

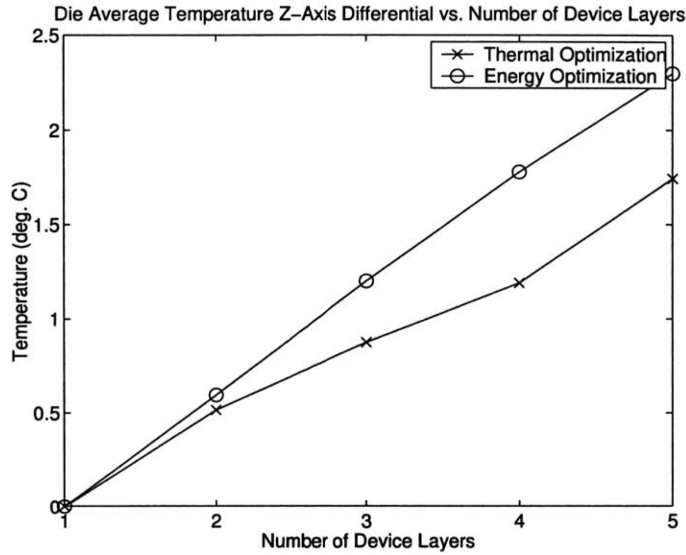


Figure 5-7: Average-temperature z-axis differential of the FFT datapath vs. number of wafers (fixed-die case).

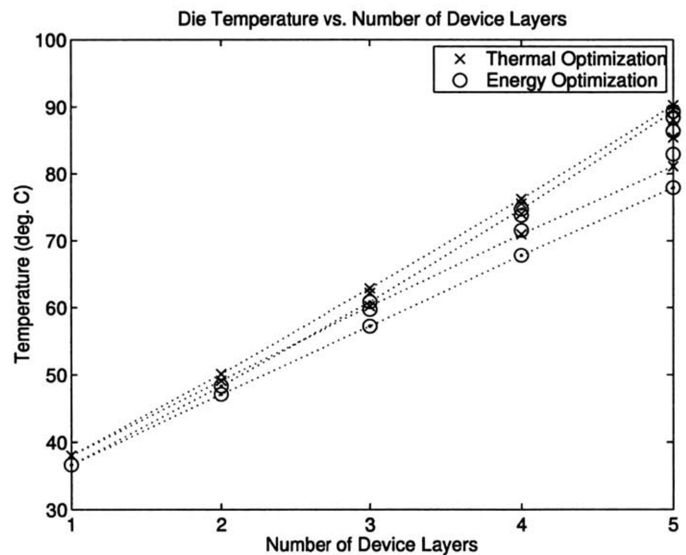


Figure 5-8: Die temperature of the FFT datapath vs. number of wafers (scaled-die case).

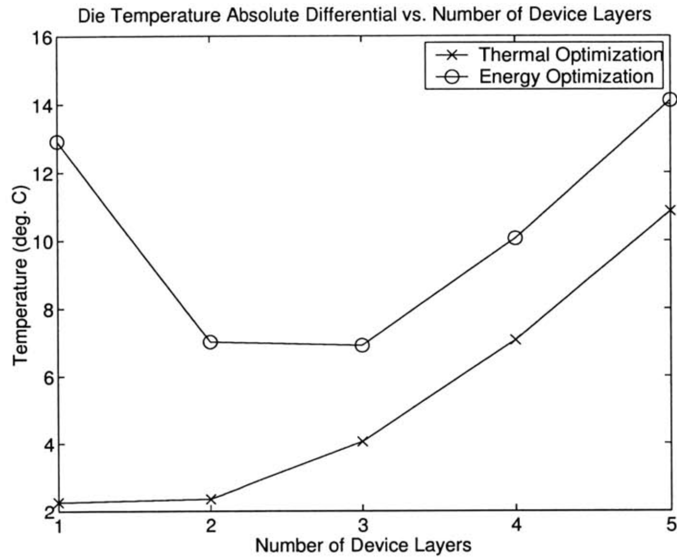


Figure 5-9: Absolute temperature differential of the FFT datapath vs. number of wafers (scaled-die case).

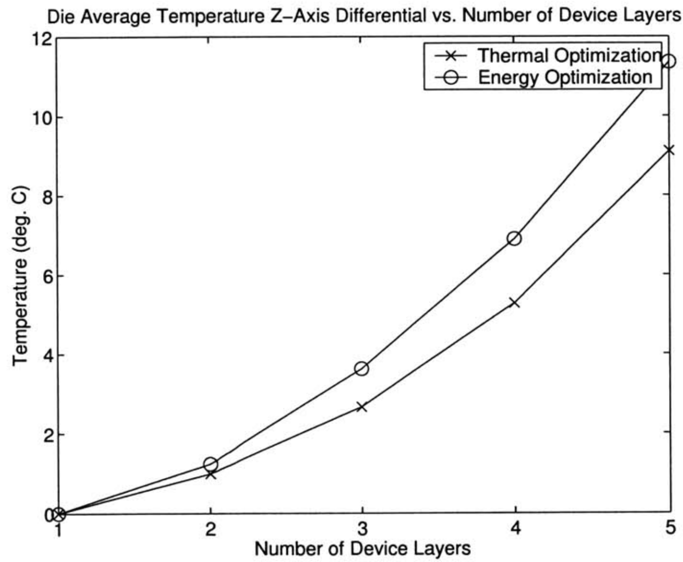


Figure 5-10: Average-temperature z-axis differential of the FFT datapath vs. number of wafers (scaled-die case).

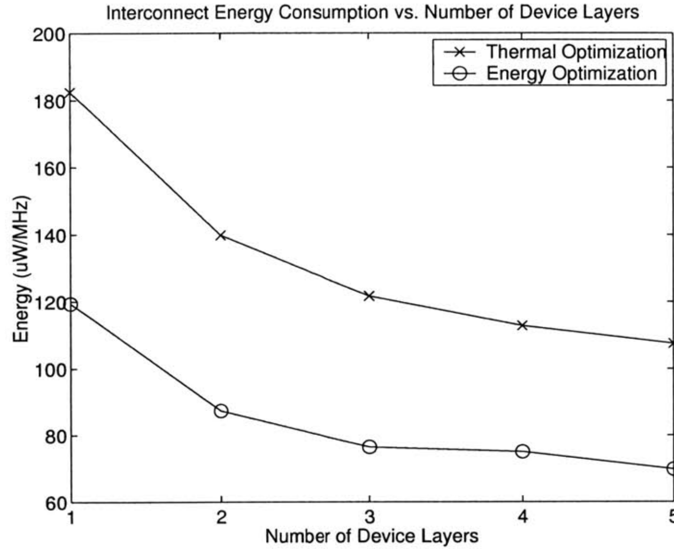


Figure 5-11: Interconnect energy dissipation of the FFT datapath vs. number of wafers in energy-optimized and gradient-optimized cases.

number of wafers used, which may be expected for general-purpose 3-D ICs.

In both fixed-footprint and scaled-footprint scenarios, we see that there is a trade-off between energy and thermal performance (which we currently construe as the most uniform thermal distribution). Specifically, we observe that the absolute temperature differential can be improved by a factor of six through the use of thermal optimization. However, the mean temperature of the thermally-optimized case is higher than that in which energy is optimized. To distribute the energy consumption uniformly, some highly-active wires must be made longer, thereby increasing energy consumption. Figure 5-11 shows that the overhead in interconnect energy dissipation when the best thermal performance is targeted is approximately 60%. Also, the graphs demonstrate that the improvement in thermal performance obtained by thermal optimization for 3-D ICs diminishes as more wafers are used and asymptotic limits are reached. Thus, the design choice of energy optimization or thermal optimization is dictated by the necessity of a smooth thermal profile (which may be the case for mixed-signal circuits or digital circuits with a severe hot-spot problem) or a lower mean temperature.

Furthermore, by comparing the fixed-die and scaled-die cases, we see that it is possible to control the die temperature through the use of extra silicon. As the energy consumption improves as the number of wafers is increased, the die area can be scaled proportionally to maintain a constant average temperature. However, if the sacrifice of silicon for thermal

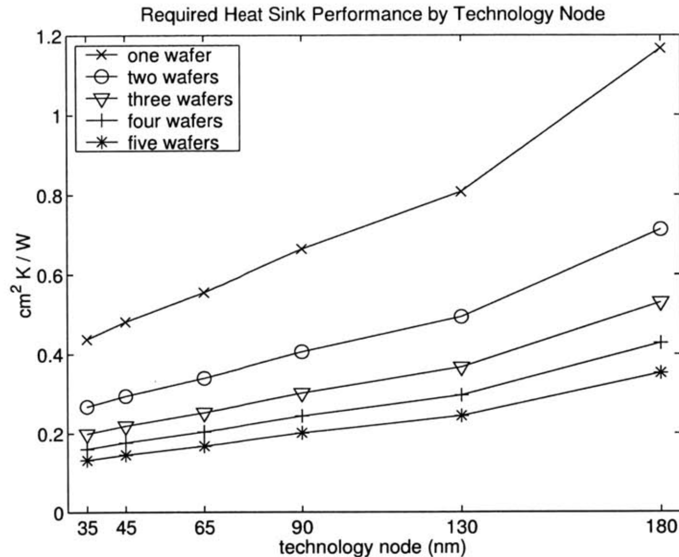


Figure 5-12: Minimum required heat sink thermal resistance by technology generation, based on ITRS projections for microprocessor size and power dissipation and 3-D performance-scaling data from this work. The desired maximum uppermost-die temperature is 100°C.

purposes is undesirable, the catastrophic thermal behavior shown in Figure 5-8 must be controlled by advanced packaging and cooling techniques.

Conversely, by using the temperature and energy performance data shown in Figures 5-5 through 5-10, we can predict the heat-sink requirements of a 3-D microprocessor with a maximum uppermost-die temperature of 100°C. Figure 5-12 shows the required heat sink thermal resistance for a microprocessor implemented in one to five wafers, for which the die area is scaled inversely with the number of wafers and the projected power dissipation scales according to the observations in Figure 5-11 and in Chapter 4. Significant improvements in conventional heat sinking or advanced cooling techniques will be required to achieve the desired scaling in circuit performance.

In the next section, we discuss one such class of advanced cooling techniques and analyze its effectiveness in allowing maximum performance scaling in three dimensions.

5.5 Active Cooling Using Microchannels

Forced fluid-flow convective transfer is a highly effective means of heat removal [114–116]. The use of microchannel fluid flow for integrated-circuit thermal management [113,117–119] has been proposed for 3-D integration in several forms [27–29]. The common goal is to

prevent the catastrophic thermal behavior shown in Figure 5-2 by providing a mechanism for lateral heat extraction.

In a microchannel scheme, a unique type of heat spreader is attached to (or fabricated on) the integrated circuit. This spreader contains embedded conduits of width 1-1000 μm through which fluid (typically water) is forced to flow. The fluid absorbs the electrical energy dissipated by the circuit as heat and carries this energy to a heat exchanger, where the energy is transferred outside the system. The fluid returns to its ambient temperature and is recirculated through the system.

Such systems may be categorized as *single-phase* (i.e. liquid or gas), in which heat is transferred principally by convection [113,118,119], or *two-phase*, in which both convection and evaporation are responsible for heat removal [117]. The combination of a heat sink and fan is an example of a conventional (macro-scale) gas-phase forced convection system. For reasons elucidated in Section 5.5.2, we consider liquid-phase microchannel systems only.

Several possible strategies exist for incorporating microchannels into a 3-D integrated circuit. Most critically, 3-D electrical interconnects and fluid-flow microchannels must be accommodated simultaneously. As stated in Chapter 2, we choose to orient the inter-wafer vias in rows parallel to the rows of standard cells; inter-wafer routing can therefore be performed without the need to allocate feed-through locations within the cell rows. This routing strategy also permits a convenient arrangement of microchannels: they may be imbedded underneath the cell rows, between the top-level metallization of a given bottom wafer and the buried oxide of the matching top wafer. The fluid flows parallel to the cell rows.

Figure 5-13 shows a cross section of a wafer-bonded structure that incorporates microchannels. Compared with Figure 1-7, this depiction is rotated 90° so that the microchannel layout can be seen. The microchannels are not depicted to scale; as Section 5.5.3 demonstrates, the actual dimensions must be engineered on a per-circuit basis. However, one evident feature in this diagram that is generally valid for all microchannel implementations is that the height of the inter-wafer vias is significantly greater with microchannels than without. This increase in height decreases the performance improvement that can be obtained and introduces another optimization trade-off for 3-D ICs.

In our analysis of microchannel cooling for 3-D ICs, we begin by developing a first-order model for die temperature that incorporates microchannels. We compare the predictions of

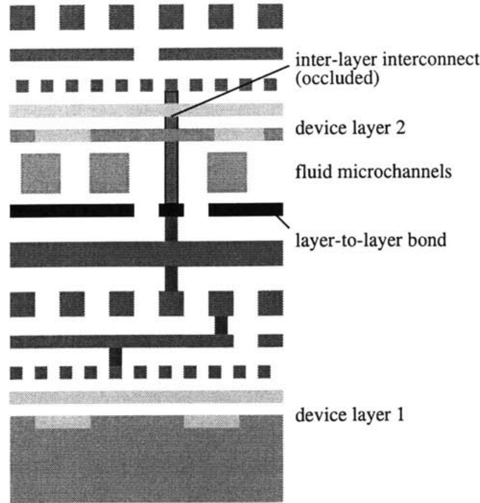


Figure 5-13: Wafer-bonded structure with the addition of fluid microchannels for cooling (c.f. Figure 1-7).

this model with a placement-based analysis of a microchannel-cooled FFT datapath. Finally, we show that microchannel implementations introduce an additional degree of freedom in designing for a desired thermal behavior. Using our model, we quantify how microchannel heat-sink design and 3-D integration combine to determine global die temperatures for a high-performance microprocessor.

5.5.1 First-Order Model

The average die temperature may be determined from first principles through the assumption that a uniform steady-state temperature T_{die} is achieved throughout the chip. We posit that in the cooling system design, N fluid-flow channels shall be distributed uniformly throughout the chip (i.e., if N_z is the number of device layers, each layer receives a cooling layer of N/N_z channels).

Figure 5-14 shows a channel model. Fluid flows in the positive x direction, and dissipated power flows into the channel according to the profile $P(x)$ (i.e. $\int P(x)dx$ over the channel length is the total power dissipated into the channel). The temperature surrounding the channel is assumed to be T_{die}, and the fluid temperature profile is given by $T_{ch}(x)$. The ambient temperature is assumed to be T_{amb} and the fluid inlet temperature is T_{in}. We assume that a change of phase does not occur; this can be verified by the examination of the resulting temperature profile. All units are assumed to be base SI units unless otherwise stated.

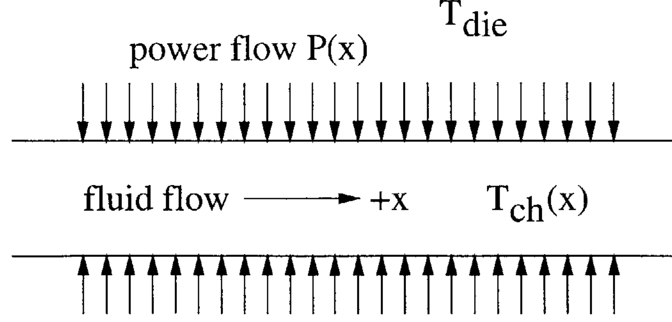


Figure 5-14: Microchannel with fluid flow in the positive x direction, power flow profile $P(x)$, and fluid temperature $T_{ch}(x)$, in an ambient solid temperature T_{die}.

Conservation of energy requires that

$$P(x)dx = A_{cs} v \rho C_p \left(T_{ch}(x + dx) - T_{ch}(x)\right), \tag{5.6}$$

where A_{cs} is the channel cross-sectional area, v is the fluid velocity, ρ is the fluid density, and C_p is the fluid specific heat capacity. An additional constraint on $P(x)$, T_{die}, and $T_{ch}(x)$ is given by the convective heat transfer relationship

$$P(x)dx = U_0 p_s dx \left(T_{die} - T_{ch}(x)\right), \tag{5.7}$$

where U_0 is the film transfer coefficient (determined by the fluid velocity, fluid properties, and channel dimensions) and $p_s dx$ is the channel sidewall area [114]. Given a square channel of cross-sectional side length s, we may combine Equations 5.6 and 5.7 into the differential equation

$$\frac{dT_{ch}}{dx} = \frac{4U_0}{sv\rho C_p}\left(T_{die} - T_{ch}(x)\right), \tag{5.8}$$

from which we determine that

$$T_{ch}(x) = T_{die} + \left(T_{in} - T_{die}\right) e^{\frac{-4U_0}{sv\rho C_p}x}. \tag{5.9}$$

To find T_{die}, we compute the total power dissipation into a single channel:

$$P_{ch} = \int_0^{x_l} P(x)dx = \left(T_{die} - T_{in}\right) s^2 v \rho C_p \left(1 - e^{\frac{-4U_0 x_l}{sv\rho C_p}}\right), \tag{5.10}$$

where x_l is the channel length. The total power dissipation is given by the sum of the

dissipations over all channels plus a residual, which we assume will exit the substrate preferentially through the bulk surface (where it is attached to the package). If P_{tot} is the total power dissipation, then

$$T_{die} = T_{amb} + R_{hs}\left(P_{tot} - NP_{ch}\right), \tag{5.11}$$

where R_{hs} is the thermal transfer resistance from the die through the die attachment to the package. Combining these equations yields

$$T_{die} = \frac{T_{amb} + R_{hs}\left[P_{tot} + NT_{in}s^2 v\rho C_p\left(1 - e^{\frac{-4U_0 x_l}{sv\rho C_p}}\right)\right]}{1 + R_{hs}Ns^2 v\rho C_p\left(1 - e^{\frac{-4U_0 x_l}{sv\rho C_p}}\right)}. \tag{5.12}$$

5.5.2 Modifications to the Thermal Algorithms

As stated in Section 5.3, the thermal-analysis engine within PR3D consists of a mesh representation of the solid substrate together with a finite-difference solver. The passive nature of thermal conduction within an ordinary IC gives rise to a finite-difference matrix that is symmetric and positive definite (SPD), which results in a matrix equation that can be solved by the conjugate-gradient method (Equation 5.5).

Conversely, the non-zero flow rate of microchannel cooling results in asymmetric heat conduction. Consider a node n inside a fluid microchannel. According to Equation 5.6, the temperature T_n of node n is determined by the temperature of the next upstream node, T_{n-1}, as well as the power flow into node $n-1$:

$$T_n = T_{n-1} + \frac{P_{up(n-1)} + P_{down(n-1)} + P_{top(n-1)} + P_{bot(n-1)}}{A_{cs}v\rho C_p}, \tag{5.13}$$

where for an arbitrary channel node i, P_i is the power flow from node i into node $n-1$, and $up(i)$, $down(i)$, $top(i)$, and $bot(i)$ are the solid-substrate nodes surrounding node i. The P_i values are determined via Equation 5.7. In other words, the dissipation of heat into a volume of fluid in a microchannel increases the steady-state temperature of the downstream fluid. More importantly, the same heat dissipation into the downstream node does not affect the upstream fluid in a symmetric manner.

The resulting finite-difference matrix is asymmetric and indefinite. As a result, the conjugate-gradient method is no longer suitable. We instead use the generalized minimum

residual method (GMRES) with ILU(0) preconditioning [120]. This method results in a longer runtime than methods suitable for non-microchannel circuit implementations with a similar number of finite-difference mesh nodes.

Accuracy and numerical stability of the solution are also of concern when microchannels are introduced. Specifically, Equations 5.6 through 5.12 are valid only in the limit as $dx \rightarrow 0$. For a microchannel modeled as a line of mesh nodes, the finite-difference approximation is invalid if the temperature increase from node to node exceeds the temperature difference between the channel fluid nodes and the surrounding solid nodes. In other words, a temperature difference ΔT between the die and the fluid results in a energy flow into the fluid and a concomitant heating of the adjacent downstream fluid in steady-state operation. However, the temperature of the adjacent downstream node is thermodynamically prohibited from rising more than ΔT. Therefore, through Equations 5.6 and 5.7, for a square channel of side length s, we find that that the separation distance l between finite-difference nodes in the channel must satisfy

$$l < \frac{sv\rho C_p}{4U_0}. \tag{5.14}$$

Within PR3D, we increase the resolution of the thermal grid along the length of the channel to satisfy this requirement.

Finally, since a die temperature consistently less than 100°C is desired, it is reasonable to simplify the model through the assumption of liquid-phase flow only. The introduction of phase change requires a piece-wise linear model that is computationally expensive. We may validate this assumption simply by checking that the maximum fluid temperature is below the boiling point.

5.5.3 Placement-Based Analysis

To assess the impact of microchannel heat-sink cooling on temperature in 3-D ICs, we analyze placements of the FFT datapath in a microchannel-equipped package. First, we design the heat-sinking system that will be used. The relevant parameters are the number of microchannels, the microchannel dimensions, and the fluid flow rate.

Choosing reasonable values for these parameters, we determine the total heat removal capacity of a given design. Our choices for the parameters are dictated by a pair of constraints. To avoid physical interference with the inter-wafer routing regions, the channel

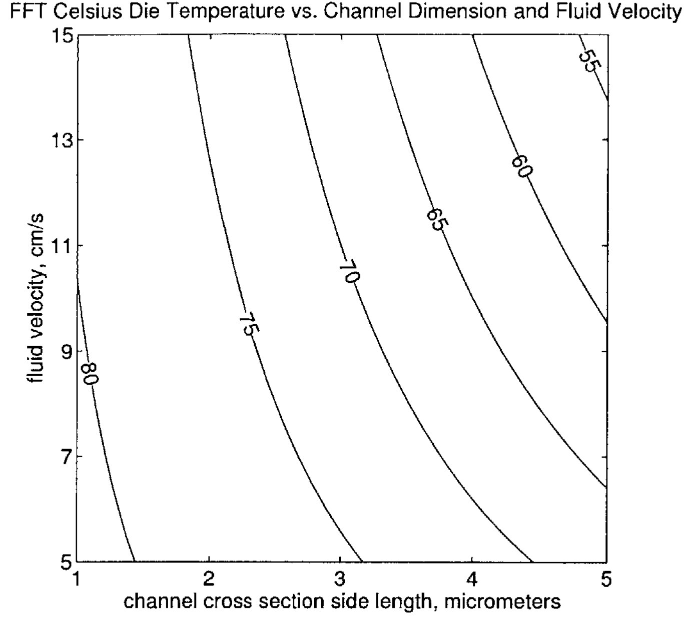


Figure 5-15: Celsius die temperature prediction for the 2-D FFT, with microchannel heat sink, as a function of channel cross-sectional dimension and fluid velocity.

effective diameter must be less than or equal to the cell height. As the standard cells used in our circuit are 5.04 µm in height, we impose an upper bound of 5 µm on the side length of our square cross-section channels. Another fundamental constraint is that friction losses in the channel result in a pressure drop along the channel length; this pressure must be supplied by the fluid source and should be as low as possible for system design simplicity. At this size scale the fluid flow is laminar for any reasonable velocity [114]; therefore, the head loss in the channel in p.s.i. is given by

$$h = 4.64 \times 10^{-3} \frac{l_{channel} v \mu}{D_i^2}, \tag{5.15}$$

where $l_{channel}$ is the length of the microchannel, v is the fluid velocity, μ is the fluid viscosity, and D_i is the effective channel diameter (which for a square channel is the side length s).

Figures 5-15 and 5-16 show the trade-off analysis for a 2-D FFT placement. Since our original analysis uses a mesh with 50 × 50 lateral dimensions, we choose to implement 24 channels in our heat sink. We also seek to minimize the cross-sectional area of the microchannels, thereby minimizing the inter-wafer via height. However, as we reduce the channel cross section, both the die temperature and the head loss increase. To achieve a reasonable head loss, we exchange cross-sectional area for decreased fluid velocity along

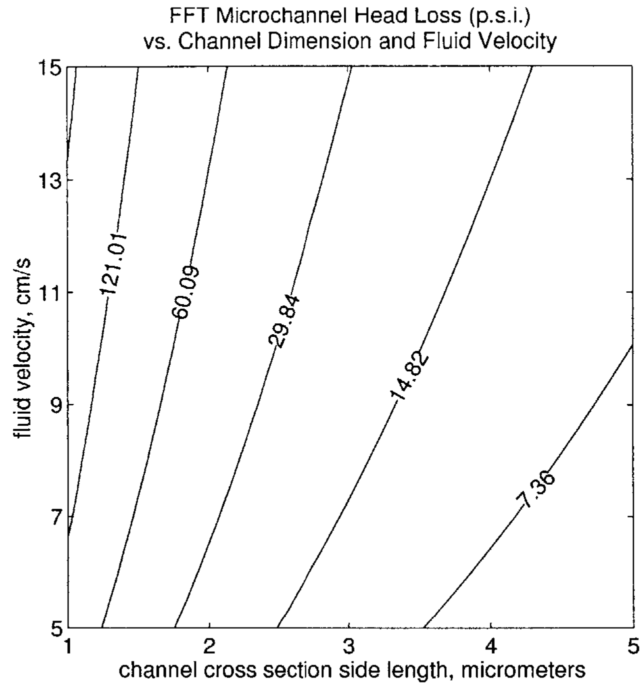


Figure 5-16: Head loss in p.s.i. for the FFT microchannels as a function of channel cross-sectional dimension and fluid velocity.

a constant-temperature curve. For the FFT circuit in particular, we opt for a 5 μm × 5 μm channel cross section with a fluid velocity of 10 cm/s. To simplify, we utilize 24 microchannels per device layer in the 3-D placements.

Figures 5-17 through 5-19 illustrate how the FFT thermal behavior changes when the number of device layers is increased. Figure 5-17 shows that the average die temperature actually decreases. As we shall explore later in this section, this is due to the linear increase in the total number of microchannels. The data indicate again that energy optimization yields the best overall die temperature. The solid lines of Figure 5-17 are the predictions of the first-order model. The model is clearly useful for predicting global thermal behavior in microchannel-cooled 3-D ICs.

Figure 5-18 shows the absolute temperature differential (maximum temperature minus minimum temperature over the entire circuit) for microchannel-cooled placements. Thermal optimization results in an approximately two-fold reduction in this differential. Figure 5-19 shows the die-to-die average temperature differential as a function of the number of wafers used. As a general trend, with a growing number of device layers, thermal optimization produces an increasingly smoother thermal profile relative to that of energy optimization.

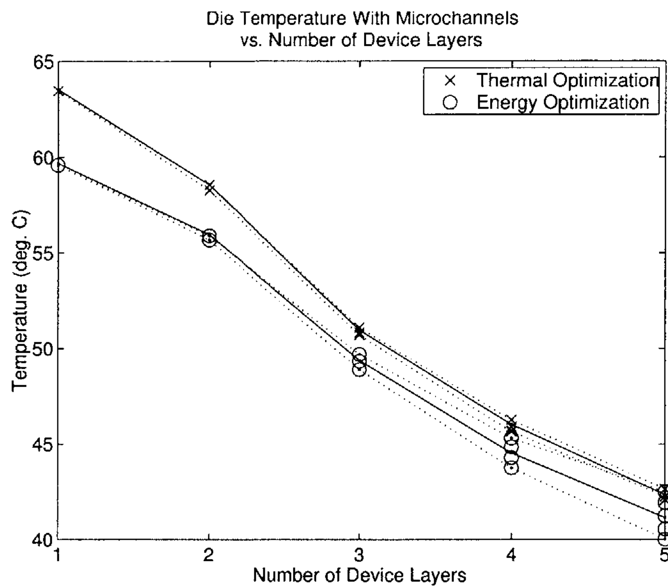


Figure 5-17: Die temperature of the FFT datapath vs. number of wafers (microchannel case).

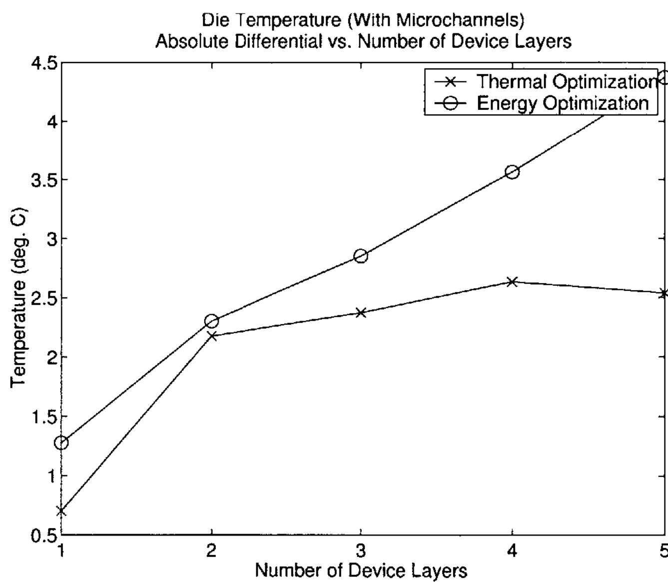


Figure 5-18: Absolute temperature differential of the FFT datapath vs. number of wafers (microchannel case).

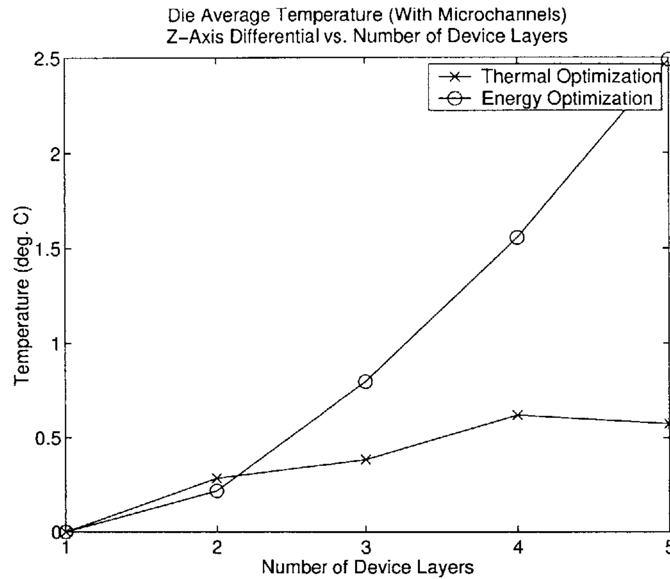


Figure 5-19: Average-temperature z-axis differential of the FFT datapath vs. number of wafers (microchannel case).

The decrease in die temperature shown in Figure 5-17 demonstrates that microchannel cooling is a powerful technology. Furthermore, the use of microchannels introduces an additional degree of freedom since the number of channels may be tuned to achieve a desired thermal performance. In Figure 5-20, we consider a high-performance microprocessor. We assume that in a conventional placement, this CPU dissipates 50 W average power over an area of 2.25 sq. cm. Using the methodology of Section 5.5.1, we design the microchannels to be 50 μm × 50 μm with a fluid velocity of 25 cm/s. Figure 5-20, generated through the use of the above model, shows how the die temperature changes as the number of microchannels and number of wafers are both varied while taking into account the 3-D power reduction data from Chapter 4. Using microchannel cooling, our choice of operating temperature is roughly independent of the choice of desired performance that we make through the number of wafers we use for integration. Microchannel heat-sink technology clearly has the potential to enable performance scaling with 3-D integration while controlling or even reversing any negative thermal side effects.

5.6 Summary

Power trends indicate that the dissipation and removal of heat will be significant issues for the design of integrated circuits. These problems are exacerbated in the context of 3-D

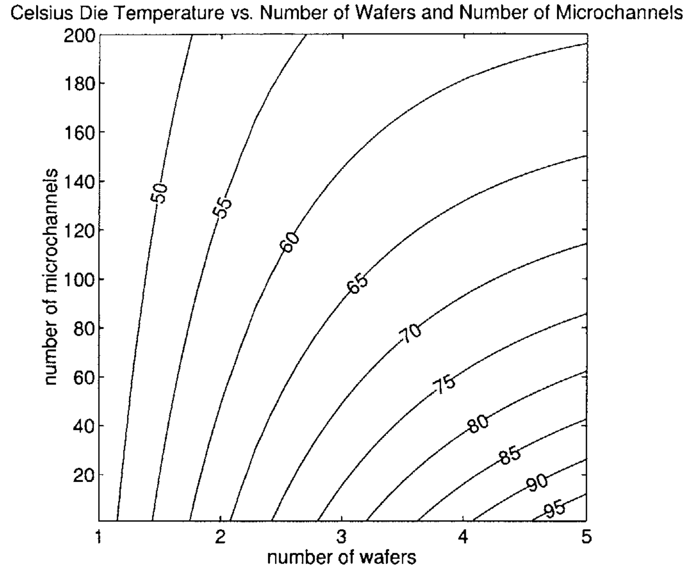


Figure 5-20: Celsius die temperature as a function of the number of wafers and the number of microchannels used. The 2-D version of this chip dissipates 50 W and has dimensions 1.5 cm × 1.5 cm. The microchannels are 50 μm in effective diameter and the water flow is 25 cm/s at 25°C at the inlet.

integration to the extent that many designers question its feasibility.

In this chapter, we presented a placement-based thermal analysis of 3-D ICs. We examined the die temperature distribution of the energy-optimized FFT datapath of Chapter 4 in two versions, one in which the die footprint is fixed (thereby sacrificing silicon for thermal purposes) and the other in which it is scaled to match the scaling of core cell area due to 3-D integration. We found that extra silicon may be used for its heat-spreading effects as one possible means of controlling the temperature of 3-D ICs. In the scaled-die form, which is considered to be more realistic from a cost-per-die perspective, we confirmed the findings of numerical models that predicted a near-linear rise in die temperature above ambient as more wafers are used for integration.

Concurrently, we examined the use of placement-based thermal optimization to control the variation of temperature within the circuit; such optimization may be needed if the overall die temperature is acceptable but the signal skew must be controlled. We found that the absolute temperature differential can be improved by a factor of six through the use of thermal optimization. However, we observed that this improvement comes at the expense of additional interconnect energy consumption, and that as more wafers are integrated into a 3-D circuit, this improvement is diminished.

Due to the catastrophic thermal behavior exhibited by the scaled-die case, we also examined the use in 3-D ICs of an advanced cooling technology known as microchannel heat sinking. We determined a first-order model for die temperature in microchannel-cooled 3-D ICs, and by utilizing the model together with placement-based analyses, we showed that overall die temperature can be regulated and even improved by employing microchannels in a 3-D IC package. At the same time, the relative improvement due to placement-based thermal optimization is preserved. Finally, we demonstrated how the cooling system and number of wafers can be selected simultaneously to obtain a desired performance level at an acceptable die temperature. With advanced cooling technologies, the performance benefits of 3-D integration can be achieved without detrimental thermal effects.

Chapter 6

Future Considerations for 3-D Integration

6.1 Overview

Previous chapters of this dissertation provided a detailed look at what can be improved in digital-system performance for circuits fabricated in current technologies. In this chapter, we explore some avenues for future work on three-dimensional integrated circuits. We look at two areas: digital 3-D IC performance in future technology generations and incorporation of digital components into mixed-signal 3-D ICs.

6.2 Predictive Technology Models: Impact of 3-D Integration in Future Technology Generations

6.2.1 Motivation

As Chapter 1 outlined, the scaling down of technology feature sizes with each generation has caused an increase in dependence upon interconnect optimization to meet system performance goals. This dependence on interconnect motivates our investigation of 3-D integration. Additionally, in the context of scaling, we desire to know how 3-D integration might improve performance in future generations.

A number of works have assessed the impact of technology scaling [4,121–124]. Table 6.1 shows device and interconnect performance for the 180 nm generation and a projected 35

node	180 nm	35 nm
V_{DD}	1.8 V	0.9 V
$\|V_T\|$	0.45 V	0.3 V
R_{wire} (mΩ/μm)	107	1760
C_{wire} (fF/μm)	0.333	0.348
wiring pitch	640 nm	120 nm

Table 6.1: Properties of devices and mid-level interconnect in 180 nm and 35 nm technologies [4].

nm node, where interconnect capacitance data was determined using typical wire-substrate and wire-wire scenarios. With this data, we can predict how 3-D ICs will perform in the 35 nm generation.

Specifically, current-generation models for the standard cells in our circuits may be scaled using the above data. For example, variations in supply and threshold voltages affect device delay as follows [125]:

$$\tau \propto \frac{V_{DD}}{(V_{DD} - V_T)^{\alpha}}. \tag{6.1}$$

By combining this effect with the scaling of transistor input and output capacitances, it has been determined that inverter fan-out-of-four (FO4) delay is roughly proportional to the drawn gate length [123]. Interconnect performance may be modeled using the same Elmore-delay methodology used in Chapter 4, if the scaling data in Table 6.1 is taken into account.

With these scaling adjustments, library files and design constraints may be produced that are appropriate for the 35 nm node. We may then obtain placements for our circuits in this projected technology.

6.2.2 Fixed-Chip Scaling

Technology scaling affords designers two benefits: more functionality may be included in chips for designers of the highest-performance circuits, and existing chips may be shrunk for an improvement in performance.

Naturally, it is difficult to obtain circuits that utilize the full functionality available at the 35 nm node. To some extent, the amount and type of desired functionality are themselves unknown. However, we can readily examine our existing circuits to see how

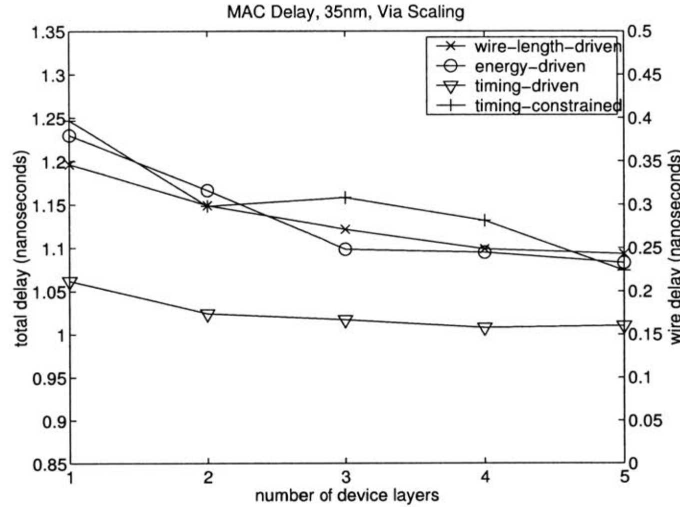


Figure 6-1: Cycle time of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling.

their performance scales. Moreover, we may also explore different scaling criteria for the strictly-3-D aspects of the technology.

Figures 6-1 through 6-4 show the performance of the 64-bit multiplier-accumulator (MAC) detailed in Chapter 4. We observe that relative to Figure 4-5 (which gives the MAC performance at the 180 nm node), gate delay scales as expected. We also note, however, that interconnect delay actually improves with scaling. The driver resistance improves as the transistor gate length decreases, and at roughly constant capacitance per unit length, the decreasing lengths due to die shrinkage result in lower wiring capacitance. Conversely, the improvement of interconnect energy dissipation at the 35 nm node behaves similarly to that shown in Figure 4-11 at the 180 nm node.

In the above analysis, we assume that the dimensions and capacitance of the inter-wafer vias scale as the gate length. For example, a five-fold improvement in wafer-bonding alignment capability is assumed. Since this capability is not a prerequisite for performance scaling in 2-D ICs, we must also consider what would happen in the absence of such scaling. Figures 6-5 through 6-8 show this analysis. We observe a performance hit of 5%-10% over the scaled-via case. Relative to overall performance gains, this is likely to be acceptable in the event that further scaling of alignment capability cannot be achieved.

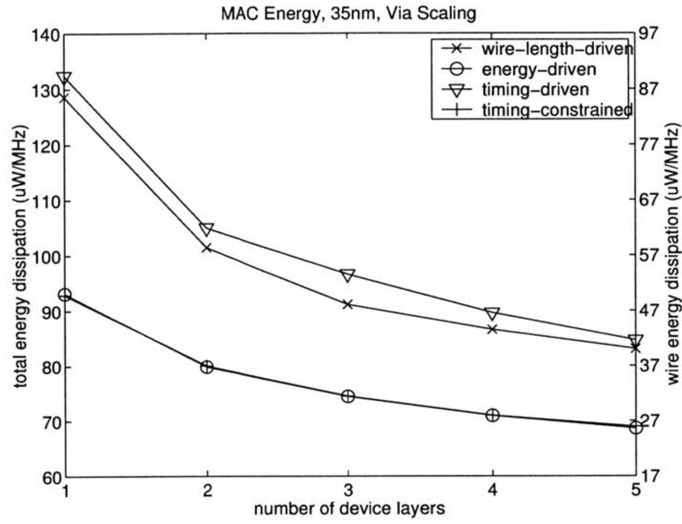


Figure 6-2: Energy consumption of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling.

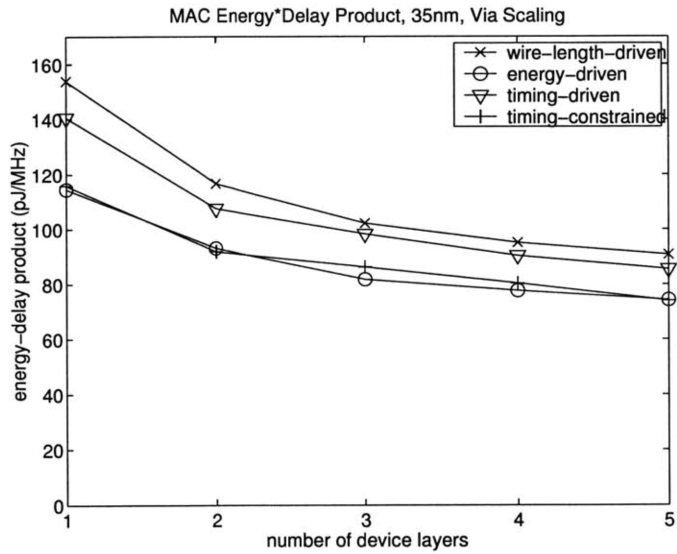


Figure 6-3: Energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling.

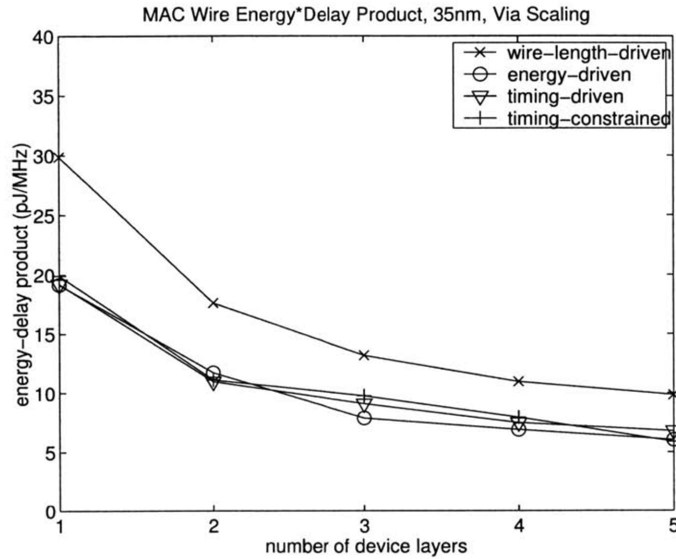


Figure 6-4: Interconnect energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology with via scaling.

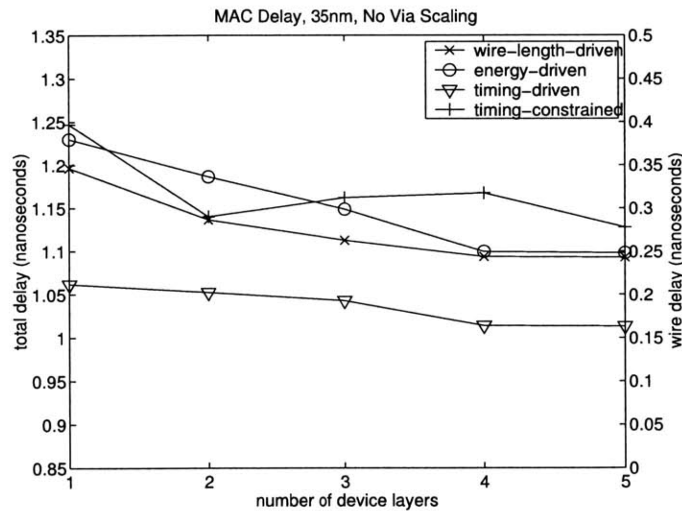


Figure 6-5: Cycle time of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling.

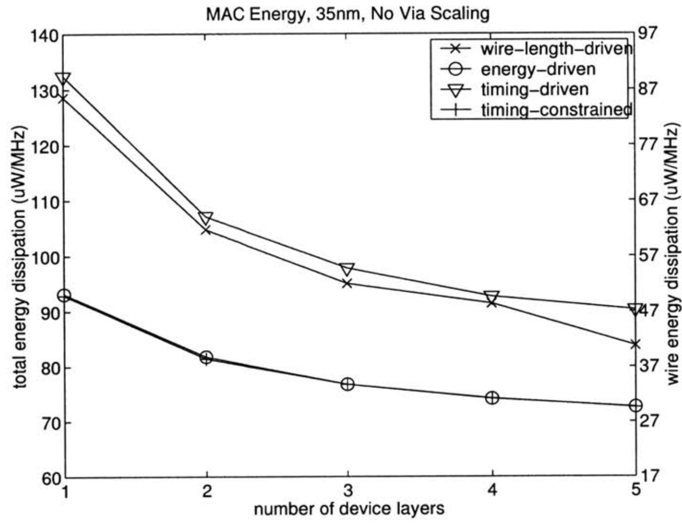


Figure 6-6: Energy consumption of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling.

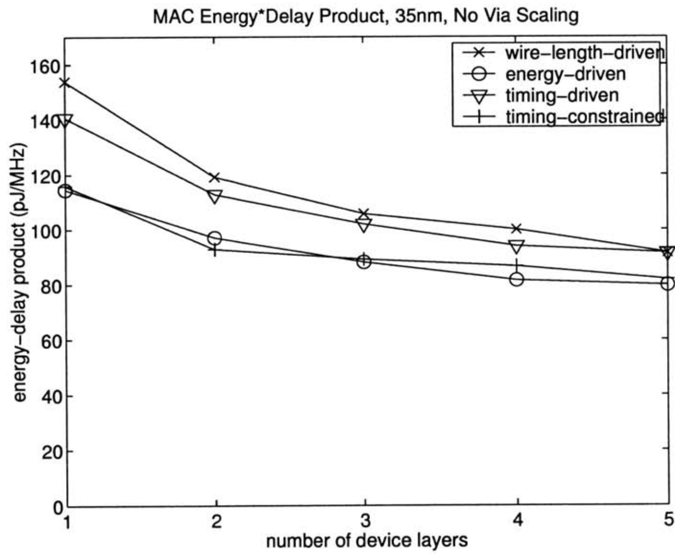


Figure 6-7: Energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling.

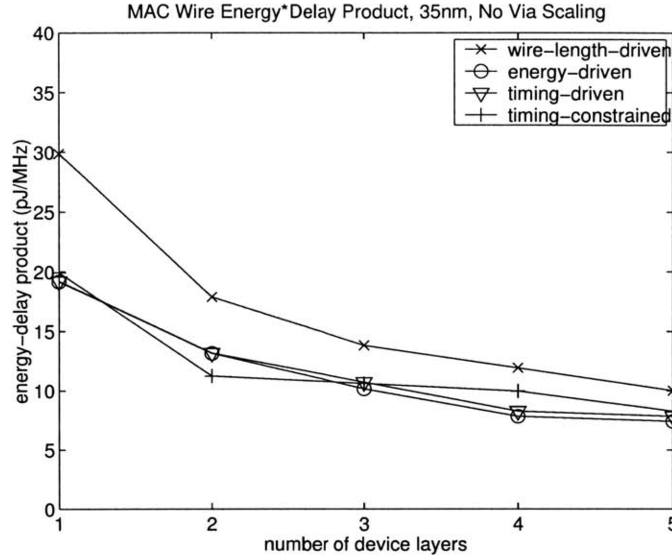


Figure 6-8: Interconnect energy-delay product of the 64-bit MAC implemented in a 35 nm 3-D technology without via scaling.

6.2.3 3-D Integration of the Projected "Largest Chip"

We note that for a given chip of fixed functionality, three-dimensional integration in future technology generations offers the same relative performance improvement, at least for energy dissipation. Thus, for these chips, 3-D integration may be considered to have an impact equivalent to some number of additional technology generations, where this number increases as the number of wafers is increased.

However, technology scaling is not truly motivated by the desire to increase performance in fixed-functionality chips. What drives performance scaling is the desire to obtain a heightened level of performance *and functionality* in the highest-end achievable circuits. Traditionally, these circuits are microprocessors.

What we actually seek to determine, therefore, is how 3-D integration might impact high-end microprocessor performance in future technology generations. Since it is infeasible to devise a new hypothetical architecture for each technology, we must satisfy ourselves with a numerical analysis. However, using the circuit data in this dissertation, we can make this analysis well-informed.

According to ITRS data [5], as well as manufacturers' stated intentions [126], the cycle time or stage delay of a modern processor such as the Pentium® 4 in terms of number of FO4 delays is decreasing from the current value of approximately 16 FO4 to an eventual minimum

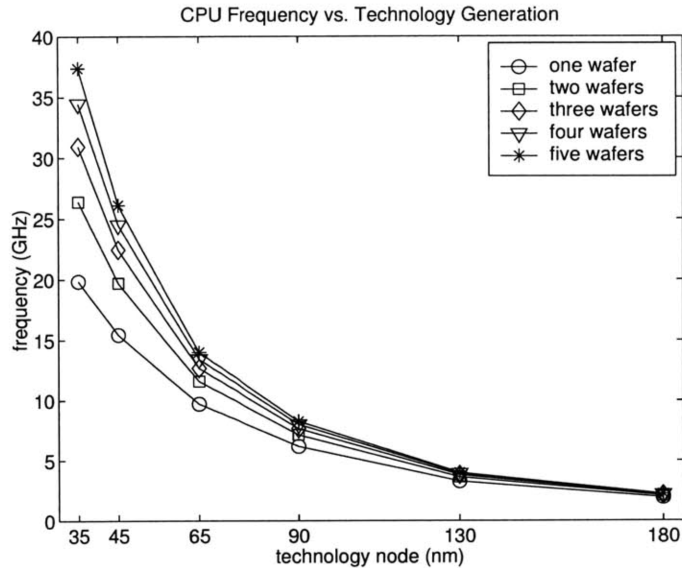


Figure 6-9: Predicted CPU frequency for several technology generations using one to five device layers for implementation.

of about 10 FO4. Concurrently, the length of the 90th-percentile wire is decreasing, due to architectural improvements, from its current value of 2 mm in the P4 (an order of magnitude less than the chip-edge length) [127].

The clock frequency of a 2-D processor implementation may be calculated for various technology nodes using this FO4 scaling data. We assume that the dominant interconnect delay component of a stage is due to driving a 90th-percentile wire in an optimally-buffered fashion. We also assume that the length of this wire scales according to the pitch scaling in Table 6.1.[1] Using these assumptions, we can predict how the interconnect delay will be affected by 3-D integration at each technology node, and thus how the clock frequency will scale correspondingly.

Figure 6-9 shows the results of this analysis. At the 180 nm node, CPU frequency can be improved by 7% to 15% by using two to five wafers. Given that modern microprocessors are carefully designed to avoid global signalling as much as possible, this is quite consistent with the results of Chapter 4. More importantly, we see that at the 35 nm node, the improvement increases to 33% by using two wafers to 88% by using five. We conclude that due to the large size scale of these circuits and the increase in dependence of total delay on interconnect delay, 3-D integration will have a significant impact in future technology

[1]This is a conservative estimate since a lesser degree of scaling results in a longer wire, resulting in a larger interconnect delay component and thus a greater overall impact for 3-D integration.

generations. Furthermore, the rate of growth of this impact exceeds that which can be achieved by conventional (2-D) technology scaling.[2]

6.3 Opportunities for Mixed-Signal 3-D Integration

6.3.1 Overview

Mixed-signal integration presents unique opportunities and challenges for circuit and system design. For memory-inclusive systems such as microprocessors, the use of additional real estate for local cache has the potential to increase performance [35]. More broadly, storage-and-processing circuits such as imagers [33] may be integrated with higher density and better performance in three dimensions than can be achieved in their 2-D counterparts.

The integration of analog circuitry with digital introduces a specific set of challenges. Digital crosstalk onto analog signals through the substrate, power and ground lines is the primary difficulty in design [128, 129]. In particular, the incorporation of high-performance digital logic with radio-frequency (RF) analog systems presents a significant problem since the digital clock or its low-order harmonics may lie in the RF tuning range.

Figure 6-10 shows the substrate noise spectrum for a 1 GHz, 1.5 V Pentium® 4 microprocessor operating under moderate load and dissipating 15 Watts. The noise induced by this level of digital activity measures 100 mV(RMS). At a peak operating power of 55 Watts, this noise increases to 190 mV(RMS). Furthermore, it has been observed that the noise voltage level increases linearly with the supply voltage and as the square root of the clock frequency [130]. This indicates that substrate noise power, like digital interconnect power, is proportional to $V_{DD}^2 f$.

Three-dimensional integration aids in the solution of these problems. For example, the bonding layer in wafer-bonding technologies, whether metal or dielectric, produces a degree of isolation [25]. Thus, as shown in Figure 6-11(a), a mixed-signal 3-D IC may be formed by designing separate wafers for the individual subsystems and bonding these wafers together.

Many of the primary issues of mixed-signal design for 3-D integration remain open. In several respects, mixed-signal monolithic-system design is the future of both 2-D and 3-D

[2]It is important to note that we have only considered the impact of 3-D integration through layout improvements to fixed architectures. It is also possible to achieve performance increases with 3-D integration by extending the architecture in ways such as by increasing the cache size and thereby reducing memory latency.

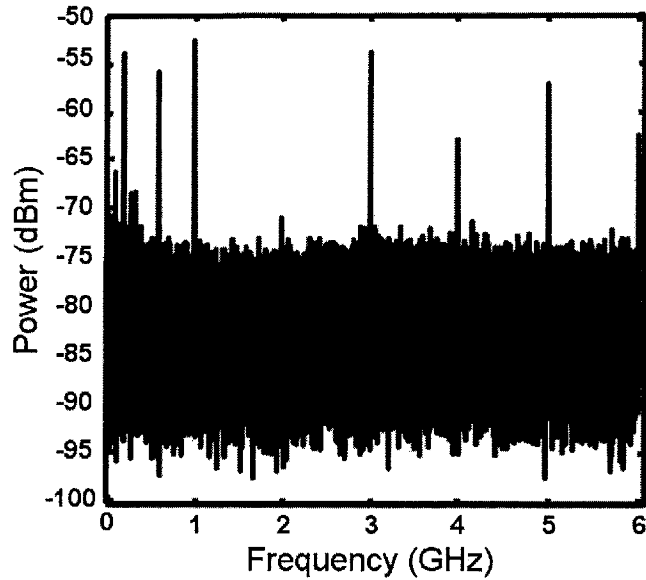


Figure 6-10: Substrate noise spectrum for a 1 GHz Pentium® 4 microprocessor operating at 1.5 V supply and dissipating 15 Watts (reprinted from [3]).

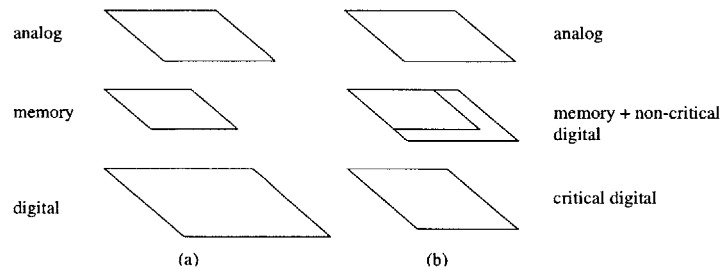


Figure 6-11: Placement of a 3-D mixed-signal system. In (a) each module is targeted for a separate wafer. In (b) non-critical digital components are placed on memory or analog wafers in order to reduce wasted silicon.

integration. To this end, in the next sections we investigate how best to optimize the digital subsystems of mixed-signal 3-D ICs.

6.3.2 Optimization for Digital Performance in Mixed-Signal Systems

Figure 6-11(a) shows the most direct method for creating a mixed-signal 3-D circuit architecture. We have already delineated some of the system advantages that this architecture achieves over a conventional system-on-a-chip (SoC). We consider the digital subsystem performance here. For example, in a single-wafer implementation of the 3-D IC in Figure 6-11(a), the digital area must be bent around the analog and memory blocks, thus increasing the corner-to-corner wire length. We expect that in addition to any analog performance improvements we obtain by using the system in Figure 6-11(a), the digital circuit can also be improved through consolidation.

However, the clear problem with this method is that the subsystem sizes will most likely be mismatched, thereby leading to the waste of silicon in 3-D implementations. Furthermore, as we showed in Chapter 4, the digital subsystem itself can be improved by using more than one wafer for integration. We therefore analyze systems such as those in Figure 6-11(b), in which suitable digital components are placed on the non-logic wafers to produce uniformity in die area.

Specifically, we examine three implementations of the 32-bit Fast-Fourier-Transform datapath and the MAC chip from Chapter 4. The first implementation is a single-chip placement in which 25% of the chip area is dedicated for a macro block (e.g. an analog subsystem) and the remainder is used for digital placement. In the second implementation, a two-wafer placement, the top wafer is dedicated for the macro block and the bottom wafer for the digital components. The third implementation is a two-wafer equal-area placement; we partition the digital subsystem so that enough of it rests alongside the macro block to produce an equal split of the whole system. Figure 6-12 shows sample layouts for these three implementations.[3]

Figures 6-13 and 6-14 show the behavior of the FFT datapath in the three implementations. We consider the third implementation twice: with small inter-wafer vias (corresponding to a via cost of 1) and with larger inter-wafer vias (corresponding to a via cost

[3]The most straightforward implementation actually is to use three wafers for the digital components and one for the macro block. However, we cannot effectively conduct any trade-off analysis with this implementation.

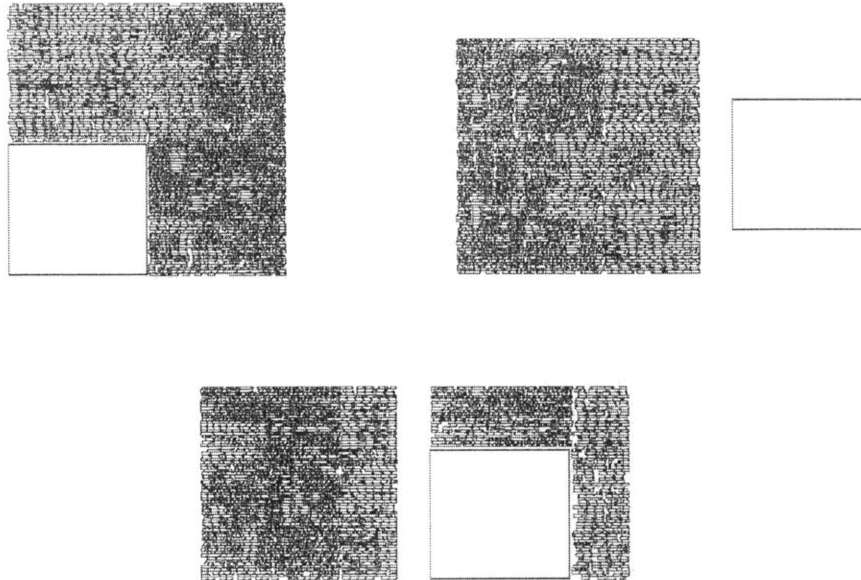


Figure 6-12: Three implementations of a mixed-signal circuit. Top left: single wafer; top right: two wafers with digital circuitry isolated to bottom wafer; bottom: two wafers with equal footprint.

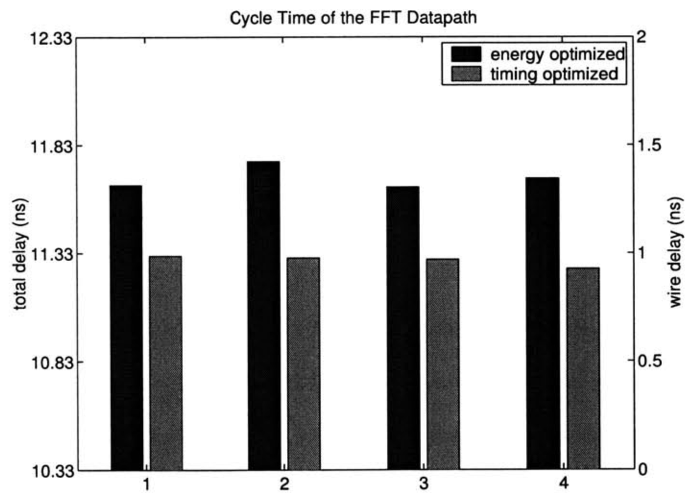


Figure 6-13: Cycle time of the FFT datapath in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias.

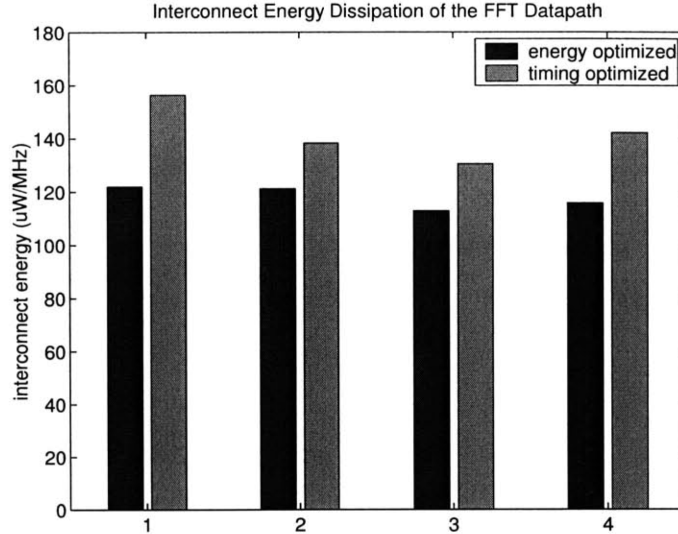


Figure 6-14: Interconnect energy dissipation of the FFT datapath in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias.

of 10). We observe no essential difference in cycle time across all four implementations. However, interconnect energy performance improves if the additional wafer is used for both digital and non-digital components (case 3). This improvement is somewhat reduced if the inter-wafer vias have larger capacitance – a trade-off since case 3 is the only instance in which these vias would be heavily used.

Figures 6-15 and 6-16 show the behavior of the MAC. Distinguishable (though not significant) cycle-time improvement is shown. However, using both wafers for digital componentry again produces a significant impact on the interconnect energy consumption.

One concern with these implementations is that they circumvent isolation, one of the main motivations for mixed-signal integration. By reintroducing digital components to the non-digital wafer, it is possible that the isolation benefit may be eliminated. The next section discusses our strategy for resolving this problem.

6.3.3 Optimization of the Digital Noise Impact on Analog/RF Subsystems

While Figure 6-10 shows that the noise floor is dictated by digital signal activity, the figure also shows that the primary source of digital noise injection is the clock signal and its

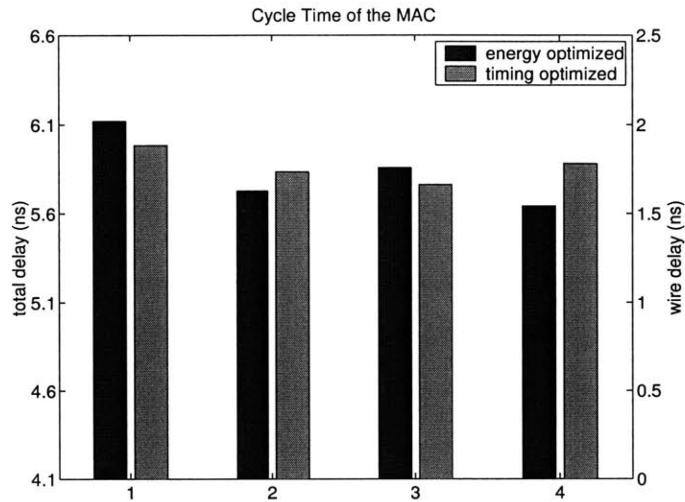


Figure 6-15: Cycle time of the 64-bit MAC in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias.

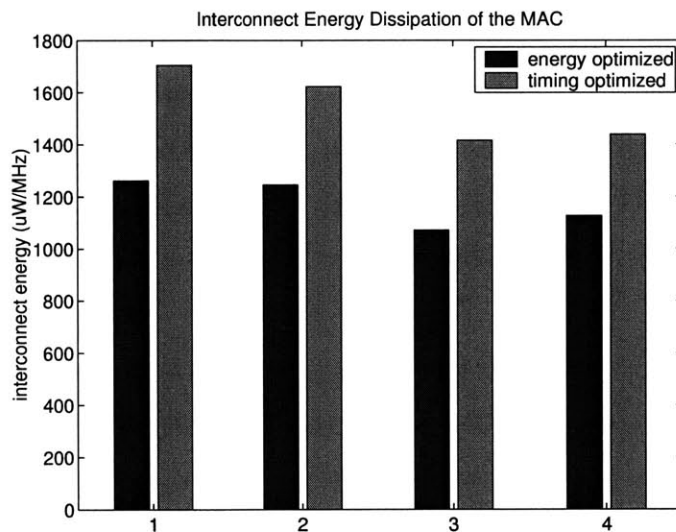


Figure 6-16: Interconnect energy dissipation of the 64-bit MAC in a mixed-signal circuit in four placement modes: (1) single-die, (2) two dice with separation of analog and digital systems, (3) two dice of equal area with excess digital on the analog die, (4) same as (3) but with larger vias.

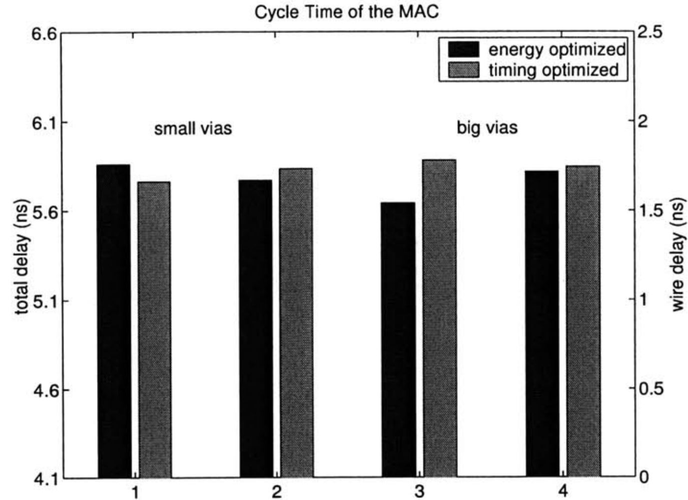


Figure 6-17: Cycle time of the 64-bit MAC two-wafer, equal-area, mixed-signal implementation. (1) and (3) are cases where the clock is distributed over both wafers; (2) and (4) are cases where the clock is restricted to the bottom wafer. (1) and (2) are cases where the inter-wafer vias are small; (3) and (4) represent larger inter-wafer vias.

harmonics. Since the clock is by definition a distributed signal, it is unlikely that digital placement-based optimization efforts will have significant impact on the amount of injected substrate noise in an RF subsystem. Furthermore, the noise introduced by signal switching typically lies outside the analog band, due to the characteristically low average switching activity [3]. However, it is possible to take advantage of these observations by restricting clock signals and their associated circuits (e.g. registers) to the digital wafer while placing some combinational logic in the excess area of the analog wafer.

Figures 6-17 and 6-18 demonstrate that essentially no reduction in digital performance exists due to restricting clock signals to one wafer. Both the interconnect energy and the cycle time remain effectively constant over the two implementations. This is reflected in the large-via case especially, as any critical paths that occupy both wafers must necessarily use two inter-wafer vias. Thus, a workable strategy for mixed-signal 3-D integration is to use non-critical digital components as fill on wafers that would otherwise waste silicon area, while simultaneously restricting the clock signals to the digital-only wafer.

As these mixed-signal systems become even increasingly complex, however, it is likely that more sophisticated CAD tools and placement strategies will be required. In the following section, we outline a possible architecture for future 3-D mixed-signal design automation.

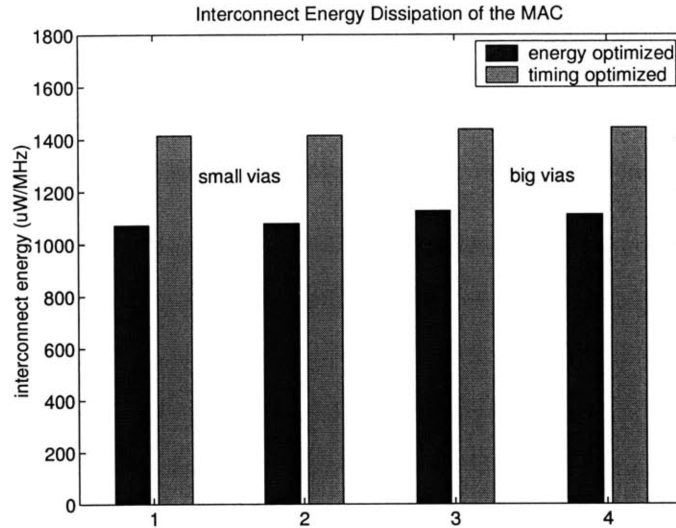


Figure 6-18: Interconnect energy dissipation of the 64-bit MAC two-wafer, equal-area, mixed-signal implementation. (1) and (3) are cases where the clock is distributed over both wafers; (2) and (4) are cases where the clock is restricted to the bottom wafer. (1) and (2) are cases where the inter-wafer vias are small; (3) and (4) represent larger inter-wafer vias.

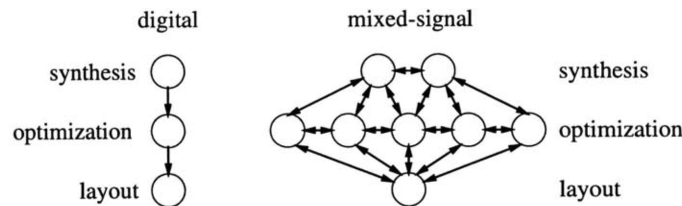


Figure 6-19: Digital vs. proposed mixed-signal design flow paradigms.

6.4 Architecture for a Design Flow for Mixed-Signal 3-D ICs

Clearly, there are many opportunities for mixed-signal circuit design for 3-D integration. Focusing solely on the digital components of such systems, we find there to be avenues for increasing performance relative to multiple-chip or even single-chip implementations.

The evident next step is to explore the automated design, synthesis, and optimization of whole mixed-signal systems. To a large extent, this remains an open problem for 2-D ICs. However, since 3-D integration is expected to enable system architectures that would be impossible in single-chip integration, a truly comprehensive 3-D design methodology cannot be developed as an extension to conventional tools (as we have done for digital circuits). Instead, a new software architecture that considers 3-D integration at all levels must be devised.

Figure 6-19 shows the paradigm shift that we envision. Instead of the linear flow ap-

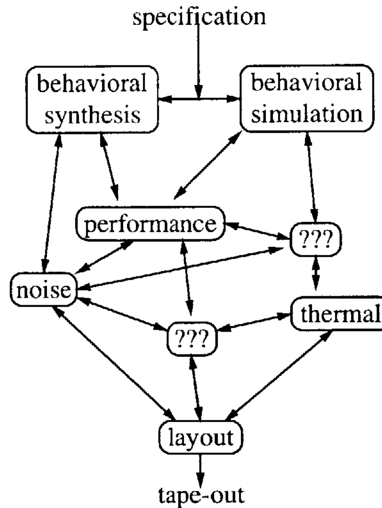


Figure 6-20: Outline of a candidate mixed-signal design flow.

propriate for digital circuits, we propose a distributed, modular tool set in which each tool has a parallel view of the entire design as it evolves. In digital-system design, it is possible to perform functional synthesis first and technology-based optimization second; the choice of function is independent of technology. However, in mixed-signal design, choices must be motivated by the options available in the technology, even at the highest levels of architecture.

For example, a system design tool may act to partition the high-level system architecture over several heterogeneous wafers. Thus, wafer-based simulation capability is required at the level of behavioral simulation. This same capability will also be required at the detailed optimization level. Due to the higher degree of redundancy when compared with digital-system design, organizing the simulation module as a cross-cutting tool that interfaces with other parts of the flow in parallel is likely to be more efficient.

Figure 6-20 proposes a candidate design flow architecture; it includes components that will be required for 3-D mixed-signal design. Again, we expect there to be some components that are unnecessary for 2-D design, as well as several for both 2-D and 3-D design that we have not anticipated. We envision three phases similar to those in the digital flow: *synthesis*, *optimization*, and *layout generation and analysis*. However, we expect that these phases will be far more inter-related.

Synthesis The digital paradigm will work to a limited extent for mixed-signal systems. Even in digital systems we see the unification of synthesis with technology-driven placement

and routing [131]. In mixed-signal design, we expect to perform some kind of synthesis at every level of the flow. For example, the initial design stage will require the development of a block architecture from a high-level specification in a language such as VHDL-AMS [132] or Verilog-AMS [133]; this block architecture may be expressed in the same language. However, at some point the blocks must be synthesized using a separate design module such as an analog/RF cell generator. The unification of top-down and bottom-up synthesis is far more likely to be required on a per-design basis than is the case with digital design. For 3-D integration in particular, different block implementations may be desired for use on different wafers.

Optimization It is less likely that one will be able to partition mixed-signal design into technology-independent and technology-dependent optimization steps (e.g. logic optimization and placement optimization, as is done with digital-system design). For example, at a high level of mixed-signal architectural specification, it may be necessary to know that the power dissipation and resulting heating of a digital subsystem will rule out some architectural options for an analog-to-digital converter (ADC). (We hypothesize that the system architecture requires some number of bits, which for a given temperature may be provided only by certain ADC architectures.) Similarly, system-level sensitivity requirements may require noise-based optimization during the high-level partitioning of a mixed-signal system, while a concurrent, low-level, noise-based optimization of the locations of individual registers and logic must also be performed. Thus, we envision a set of modules that are topical masters, such as noise-based and thermal optimizers, rather than a sequence of modules that are flow masters, such as a logic synthesizer followed by a placement engine.

Layout Generation and Analysis In mixed-signal design, as with digital design, the final goal is also fully-developed layout. The generation of layout must include digitally-motivated optimization techniques (such as performance-driven routing) and mixed-signal considerations (such as digital-to-analog coupling). Other issues such as the impact of lower-wafer wire self-heating on upper-wafer substrate temperature must be considered. Additionally, in keeping with the task-master paradigm we espouse for mixed-signal design automation, the layout engine should provide detailed feedback to other engines such as simulation and performance modules.

To develop this architecture into a usable flow of tools, it is clear that several innovations will be necessary. However, the need for such a design flow is undisputed. Existing tools do not even begin to approach the level of functionality required for such design.

6.5 Summary

Our work in this dissertation illuminates how three-dimensional integration can be used to improve system performance in digital ICs. In this chapter, we have examined how these improvements can be sustained over the entire lifetime of 3-D integration technology and VLSI technology in general.

We began with a study of 3-D integration in the context of technology scaling. We found that for circuits of fixed functionality, 3-D integration has equal impact on energy dissipation across all technology generations. Furthermore, for small circuits, the impact on cycle time is diminished due to the simultaneous decrease in both driver resistance and wire capacitance. This would suggest that at best, 3-D integration will keep pace with technology scaling, and that we might even think of 3-D integration as equivalent to some number of additional generations.

However, by extending this analysis to microprocessors, we found that 3-D integration could enable performance increases at a rate above and beyond that achievable with technology scaling. In these circuits, increases in wire delays due to technology scaling result in cycle times exceedingly dominated by these delays. Thus, when integrated using multiple device layers, the performance of these circuits improves more drastically in future technology nodes than in current nodes. For example, a CPU fabricated at the 180 nm node exhibits a modest 15% clock-frequency increase when integrated in five wafers, whereas a CPU at 35 nm could be accelerated by 88% by using five wafers.

Having examined this aspect of the future of 3-D integration, we turned to mixed-signal design. We considered some methods for the 3-D integration of logic circuitry with non-logic macro blocks such as analog, memory, and MEMS technologies. While the most straightforward approach for such integration is simply to fabricate these subcircuits on separate wafers and then bond them together, we determined that digital circuitry can be placed in unused areas on the non-logic wafers without detriment to the non-logic components. In particular, we tested strategies for restricting digital clock signals such that analog and dig-

ital components could rest side-by-side without introducing unwanted noise into the analog components.

Finally, we devised a hypothetical path for extending our design-tool framework to incorporate mixed-signal automation for 3-D ICs. We conjectured that the most difficult challenge to mixed-signal design automation lies in developing the ability to integrate different performance aspects such as delay, energy, noise, and temperature across all components and stages of design. Furthermore, with 3-D integration, the performance analysis of mixed-signal circuits is not algorithmically the same as in conventional mixed-signal circuits, due to the lack of useful abstractions. Thus, new techniques and paradigms must be created to solve this important problem.

Chapter 7

Conclusion

7.1 Summary of Research Results

In this dissertation, we have examined an emerging technology called three-dimensional integration, which we have defined to be any technology in which multiple planes of active devices can be wired together by an electromechanical interconnect. We have made two fundamental contributions: a set of computer-aided design tools for the construction of 3-D ICs and the performance analysis of these circuits.

The tool set consists of two programs. The first is PR3D, a placement and global routing tool we have developed for 3-D ICs. Given the number of device layers and parasitic cost associated with an inter-layer wire, PR3D can perform a placement-based optimization of circuit performance. It considers metrics such as wire length, energy, timing, and thermal characteristics. The second tool is 3-D Magic, a layout editor. Through its user interface, 3-D Magic features design-management additions that could not be obtained in prior methodology-based design flows. 3-D Magic also provides layout-versus-schematic, design-rule checking, and parasitic-extraction capabilities for 3-D ICs.

Concerning performance improvement in 3-D ICs, the body of prior work had revealed an enormous potential. Our work has quantified this potential through the use of actual circuit placement and simulation. We have verified the wire-length predictions of previous work and shown that this predictive capability is accurate to within 20% of placement and routing. We have found, using 3-D placement and routing, that the total wire length of a given standard-cell circuit may be reduced by 27% to 51% by using two to five device layers. We have determined that the length of the longest wire in a circuit may be reduced

by 31% to 56%, again by using two to five wafers. We have also extended this analysis to consider the impact of inter-layer via dimensions on these wire-length figures, and found that while the total wire length is strongly affected by these dimensions, the longest wire is not similarly affected.

With the use of circuit-based analyses, we have been able to make the first specific determinations regarding more important metrics such as energy dissipation and cycle time. By considering three circuits, we have found that with the use of five wafers, we could obtain up to a 54% reduction in wire delay, 54% reduction in interconnect energy dissipation, and 75% reduction in wire energy-delay product. Furthermore, we observed greater performance improvements in the larger chips.

These discussions of performance motivated an additional analysis of heat dissipation, a potentially problematic circuit issue in 3-D ICs. Building on numerical analyses of thermal performance, we have carried out a placement-based thermal analysis of 3-D ICs. In so doing, we were able to quantify the trade-offs associated with optimizing for best thermal performance versus best energy performance in a 3-D circuit. We found that, for example, up to a factor of six improvement in thermal gradient could be obtained, but at a cost of up to a 60% increase in interconnect energy consumption. Furthermore, we determined that while the percentage energy overhead remained relatively constant as we increased the number of wafers, the benefit of thermal optimization tended to decline.

We have also confirmed that the overall thermal outlook for 3-D integration is bleak with conventional packaging approaches. However, we have analyzed two solutions for this problem. By considering a fixed-die approach, in which the 2-D form factor is held constant while the number of wafers is scaled, we have shown that excess silicon could be used for heat-spreading purposes, thus maintaining an acceptable die temperature. This comes at an additional manufacturing cost, so we have also explored the use of microchannel fluid flow for cooling 3-D ICs. We have devised a numerical model for die temperature in a microchannel-cooled circuit and confirmed the behavior of this model using placement-based simulation. With the use of advanced packaging solutions, we have shown that temperature in 3-D ICs can in fact be controlled.

Our final results in 3-D integration examined the role of 3-D integration in future technologies. We considered how 3-D integration would improve performance in future technology generations. For chips of fixed functionality, we have found that the improve-

ment due to 3-D integration is preserved across generations. We have determined, however, that for high-end circuits such as microprocessors, the impact of 3-D integration will be even greater in future technologies than it is currently. A modest 15% improvement gained by using five wafers at the 180 nm node becomes an 88% increase at 35 nm. We have also examined ways of extending our work into the mixed-signal domain. We have considered a few different approaches for optimizing the digital subcomponents of a mixed-signal circuit, and found that it is possible to restrict sources of noise (such as the clock) from analog subcomponents while still maintaining the digital-system performance.

7.2 Directions for Future Work

The results summarized in the previous section provide several avenues for further research. We discuss these in the categories of technology, CAD, and circuit design.

7.2.1 Technology Research

Our work in Chapters 3 and 4 has made it clear that the performance of inter-layer interconnect is critical to digital 3-D system performance. Two important aspects to this interconnect exist: feature size and parasitic performance.

It is critical both for the performance of current architectures and invention of new ones that a high-density interconnect be available. Specifically, we postulate that for new architectures to be devised, the inter-layer interconnect pitch must be within an order of magnitude of the minimum feature size. This will allow the use of inter-wafer wires for local or semi-local interconnect. Similarly, the capacitance (and in future technology generations, inductance) of these interconnects must be controlled, such that future 3-D performance improvements are not ameliorated.

Thus, continual research into bonding alignment strategies is a necessity. Alternatively, we can find suitable bonding approaches that avoid the need for high-precision wafer alignment.

In addition, our research into the thermal properties of 3-D ICs has shown that advanced cooling technologies will be highly beneficial, if not absolutely required. Further research into useful ways for integrating this technology into the wafer-bonding process is merited. Furthermore, the potential for integrating micro-electromechanical (MEMS)

technology with circuits in a 3-D fashion includes the possibility of incorporating other MEMS-style heat-removal mechanisms into the 3-D structure. This general area represents one of the highest priorities for continued research into three-dimensional integration technology.

Finally, yield analysis and optimization is also an important problem for several kinds of 3-D integration. In a wafer-bonding technology, for example, there are two issues. First, the commercial viability of the technology is dependent on a reliable bonding mechanism. If 3-D integration becomes the primary yield bottleneck, its use will at best be confined to expensive, high-performance flagship components. Second, if a reliable, high-quality bonding procedure cannot be achieved, new verification and test procedures will have to be developed to ensure that all the inter-wafer interconnects are formed correctly.

7.2.2 CAD Tools

As we have described in Chapter 6, we envision that future design tools will be built on architectures quite different from those in use today. In the digital domain, the linear flow of tools is already being reorganized into tool sets in which early stages such as synthesis are merged with global-routing-driven predictors of final layout.

Several second-order digital optimization problems in 3-D integration have yet to be solved. Optimizations in the space of 3-D detailed routing likely exist that are not covered by our present approach. Furthermore, the simultaneous consideration of electrical and thermal performance during routing is a potential opportunity. Inter-layer interconnects may be introduced, for example, that serve not to carry signals, but to carry heat.

Another area for further study is in design for testability. If, as discussed in Section 7.2.1, 3-D integration technology becomes a major yield bottleneck, a mechanism for testing individual device layers before bonding must be devised. Alternatively, the entire existing stack of a partial 3-D IC may be tested prior to integration of the next layer of the stack. In a conventional IC, a *scan chain* is designed into the circuit whereby a test pattern may be loaded into the registers. The circuit is then allowed to operate for one or more cycles, and the resulting register values are scanned out of the circuit. In an unfinished 3-D IC, some bypass mechanism might be devised so that only the inter-wafer interconnects are tested; logic functionality may be verified using the full scan chain when the circuit has been completed.

We also posit that computer-aided design for mixed-signal circuits, both 2-D and 3-D, will be an important field of research. There continues to be growth in the demand for system complexity in a way that can only be satisfied by system-on-a-chip integration. At the same time, many integration constraints can only be met by new technologies such as 3-D integration. Furthermore, the capabilities of design tools for this sort of mixed-signal system design lag well behind those of digital design tools. Analog design has long been considered an art impervious to the kind of systematization required for computer-based automation and optimization.

Several specific problems require addressing. For example, the development of a flexible cell-based analog synthesis tool is an ongoing task of key importance. Also, a fast substrate-noise prediction or analysis tool for use in iterative placement methods will be required.

7.2.3 Circuit Design

The results we have produced regarding the performance of digital ICs present a strong case for continued research into 3-D integration technology. However, one facet that we have not explored is the use of 3-D integration for improving or replacing specific circuit architectures. The volumetric scalability of integration in three dimensions may be exploited by various forms of communication-centric chips.

Field-programmable gate arrays (FPGAs) have already been investigated as a candidate architecture for scaling in three dimensions. However, much more investigation is warranted. Moreover, the availability of mixed-signal integration leads us to conjecture that some form of programmable mixed-signal fabric may be possible. Other computational fabrics such as distributed multiprocessor-memory systems will likely also scale in performance if implemented in a 3-D IC.

Other architectures certainly exist, some as yet unconceived, that will leverage 3-D integration for even greater performance gains than have been demonstrated in this dissertation. The full extent of such gains can only be determined by further research. We therefore expect that with innovative thinking, three-dimensional integration will truly flourish.

Appendix A

Usage Information for the 3-D Design Tools

A.1 PR3D: The Placement and Routing Tool

A.1.1 Platform Support

PR3D is written in C with the intent to be portable to any platform supported by the GNU C compiler (gcc) and associated build environment. PR3D has been built and tested successfully on i386 and Alpha Linux platforms as well as the Sun Solaris environment.

A.1.2 Usage

Overview

PR3D is invoked as a batch-processing tool. The relevant input files and output options are passed via the command line. The main input is an auxiliary (`.aux`) file that specifies the floorplanning or placement data:

```
PR3D -f input.aux
```

The auxiliary file uses an extension of the GSRC format, and thus contains a single line:

```
<problemtype> : file_1 file_2 ... file_n
```

where `<problemtype>` is one of `RowBasedPlacement`, `Routing`, `LEFDEF`, or `LEFDEFrouting`. The file list for the GSRC placement and routing problems contains the `.nodes`, `.nets`,

.wts, .scl, and .pl files, while the LEF/DEF problems require the library .LEF, design .DEF (for placement) or .3DDEF (for routing – more on this in the section on output formats), and optionally the design switching activity .SAIF, timing constraint information .SDF, and cell timing library .TLF.

A typical invocation thus might be:

```
PR3D -f input.aux -n 3 -z 10 -c -y -s output.def
```

Specific optimization and output modes are covered in the next section.

Command-Line Arguments

Here are the command-line options for PR3D.

```
   option                      argument          default argument/action
--------------------------------------------------------------------------
   -f|--auxFile                <filename>        [no default]
   -i|--partitioner            HMETIS|PATOH      [built-in partitioner]

   -n|--numStrata              int >= 1          [1]
   -p|--part3DFirst            [no argument]     [disabled]
   -z|--zAxisScale             int >= 1          [1]

   -r|--saveRentData           <filename>        [no default]
   -s|--savePlacement          <filename>        [no default]
   -t|--savePlacementStats     <filename>        [no default]
   -w|--saveWLdist             <filename>        [no default]

   -c|--constrainTiming        [no argument]     [disabled]
   -o|--optimizeTiming         [no argument]     [disabled]
   -y|--useSwitchingActivity   [no argument]     [disabled]

   -g|--thermalGrid            <filename>        [no default]
   -k|--constrainThermal       [no argument]     [disabled]

   -h|--help                   [no argument]     [prints help message]
```

Here is a description of the various options.

Global options:

- `-f|--auxFile`: see above.
- `-i|--partitioner`: PR3D can utilize a built-in multi-level partitioning code or use the the `hMetis` [52] or `PaToH` [54] libraries available from the WWW.

- `-h|--help`: provides a help message.

3-D optimization options:

- `-n|--numStrata`: (e.g. `-n 5`) specifies the number of device layers.
- `-p|--part3DFirst`: if specified, tells PR3D to partition into device layers before any other partitioning.
- `-z|--zAxisScale`: specifies the inter-layer via cost, as defined in Chapter 3.

Output options:

- `-r|--saveRentData`: saves pin-versus-block data to the specified filename.
- `-s|--savePlacement`: saves the placement to the specified filename, using the format corresponding to the input data.
- `-t|--savePlacementStats`: saves the wire lengths of the individual wires to the specified filename.
- `-w|--saveWLdist`: saves the wire-length distribution, in histogram format, to the specified filename.

Optimization options:

- `-c|--constrainTiming`: constrains the timing of the placement to the delay specified in the `.SDF` input.
- `-o|--optimizeTiming`: optimizes the cycle time of the placement.
- `-y|--useSwitchingActivity`: optimize the placed wire lengths according to the switching activity in the `.SAIF` input.

Thermal options:

- `-g|--thermalGrid`: compute the substrate temperature based on a thermal grid analysis using material properties specified in `filename`.
- `-k|--constrainThermal`: optimize the placement for smoothest thermal distribution using the preceding grid (must be specified with `-g`).

A.1.3 File Formats

PR3D is designed to be a drop-in replacement for a conventional place-and-route tool such as Cadence® Silicon Ensemble®. As such, it supports the LEF and DEF file formats [134] as well as the GSRC bookshelf format [135].

Output Formats

Conventional placement formats are quite clearly not suited for 3-D placement. The GSRC `.pl` placement format is, however, easily extended to three dimensions. The format consists of lines of the form

```
<cell name>      x   y   :  <orientation>   [: optional extensions]
```

The device layer is merely included in the optional extension list.

The more industrial `.DEF` format requires some additional token support for 3-D IC placement. In what we have named the `.3DDEF` file format, the global keyword `STRATA` is used to indicate the number of device layers in the placement. In the `COMPONENTS` and `NETS` sections, the `+ STRATUM` modifier fixes layout components to specific device layers.

Configuration File Formats

The main configuration files are the `.aux` file, described above, and the thermal configuration associated with the `--thermalGrid` option. A sample configuration is given here. Essentially, the solid material properties are given in the first section and the breakdown of material layers for each device layer is given in the second. `scaling scaled` indicates that in 3-D the die size is to be scaled inversely with the number of device layers; the alternative is `scaling fixed`, for which the 2-D die footprint is used in all cases. In the `layers` section, the materials for the repeatable layers are given as mixtures, where `x` represents a homogeneous mixture, as opposed to `|` and `-`, which represent vertical and horizontal interleaved stripes, respectively.

```
begin solids
# material thermal conductivities in W / m K
# material          x            y           vertical(z)
  Cu                392          392         392
  Si                145.7        145.7       145.7
  SOI_Si            70           70          70
  SiO2              0.6          0.6         0.6
end solids

begin layers
  scaling scaled

  begin fixed
```

```
# fixed layers are given in W / m K
#   name          conductance (W / m K)      thickness (nm)
#                 x       y       z
    heatsink      0.002   0.002   0.002        1000
    substrate     145.7   145.7   145.7      499000
  end fixed

# the rest are repeatable layers, i.e. list here all the layers
# that belong to a stratum, in order.
  begin stratum
#   name          material                  thickness (nm)
    substrate     SOI_Si:0.25xSiO2:0.75     1000
    dielectric    Cu:0.01xSiO2:0.99         1000
    metal1        Cu:0.3-SiO2:0.7           1000
    dielectric    Cu:0.01xSiO2:0.99         1000
    metal2        Cu:0.3|SiO2:0.7           1000
    dielectric    Cu:0.01xSiO2:0.99         1000
    metal3        Cu:0.2-SiO2:0.8           1000
    dielectric    Cu:0.01xSiO2:0.99         1000
    bonding       Cu:0.5xSiO2:0.5            600
    BOX           Cu:0.01xSiO2:0.99         1000
  end stratum
end layers
```

A.2 3-D Magic: The Layout Editor

A.2.1 Platform Support

3-D Magic, as an extension of Magic, is supported on all platforms on which Magic is supported. 3-D Magic is based on Magic release 7.1.

A.2.2 Usage

3-D Magic is invoked from the command line by using a 3-D-augmented technology file, e.g.

```
magic -T tech_3D.tech27
```

Once started, 3-D Magic will open a layout window and provide a command-line interface for user input. In 3-D Magic, the title bars of the windows are augmented to give some information about the 3-D stack. Each window title tells which device layer is being edited and what device layers are bonded to it (if any). If the bond is face-to-face, the title will say "flipped."

3-D vias (i.e. those named in the `contact3D` section described below) may be painted in the same way as traditional vias. However, if the design in which the 3-D via is painted is bonded to another design, 3-D Magic will automatically paint a hint region on the corresponding wafer.

A.2.3 Commands

We have added several commands and subcommands to 3-D Magic.

- `:bond` bonds the edit cell to another cell. If the other cell does not exist, it is created. Syntax:

  ```
  :bond cellname top|bottom [flipped]
  ```

  `top` signifies the edit cell's metallization; `bottom` signifies its substrate. If `flipped` is specified, the bond is face-to-face or back-to-back. Otherwise, it is face-to-back.

  ```
  :bond show top|bottom
  ```

  tells 3-D Magic to name the cell bonded to the top or bottom of the edit cell.

- `:unbond` removes the bond from the specified side. Syntax:

  ```
  :unbond top|bottom
  ```

- `:select` works precisely as in Magic; however, in 3-D Magic, if a wire spans multiple wafers, all electrically-connected material on all wafers is selected. Similarly, if a subcell is bonded to another subcell within another design, selecting the cell will also select bonded cells.

- `:extract` has been extended to incorporate the subcommand

  ```
  :extract stack
  ```

  which extracts the entire 3-D stack to a single `.ext` file. For example, if three designs, `wafer1`, `wafer2`, and `wafer3`, are bonded together, then `:extract stack` will produce `wafer1+wafer2+wafer3.ext`.

A.2.4 Extensions to the Magic Technology File Format

To support 3-D IC design, two extensions to the Magic technology file format have been made. First, we have added a section akin to `contact` called `contact3D`. In this section, any types in the `types` section may be designated as inter-wafer contacts. For example, if `cutop` is defined as the bonding metallization, then the line

```
cutop top
```

indicates that for cells that have other cells bonded top-side, any painted `cutop` should be hinted on the bonded cell.

Second, we have added a design rule, `exact_overlap_3D`, to the `drc` section. This rule has the syntax

```
exact_overlap_3D  paint1,paint2,paint3,...
```

The listed paints must be in the `contact3D` section; if this design rule is invoked, any such paint in a bonded cell must be exactly aligned with an identical region of appropriate paint on the matching bonded cell.

Bibliography

[1] http://www.shef.ac.uk/eee/esg/packaging/trimod.html.

[2] N. Koshoubu, S. Ishizawa, H. Tsunetsugu, and H. Takahara. Advanced flip chip bonding techniques using transferred microsolder bumps. *IEEE Trans. Comp. Pack. Tech.*, 23(2):399–404, 2000.

[3] L. M. Franca-Neto, P. Pardy, M. P. Ly, R. Rangel, S. Suthar, T. Syed, B. Bloechel, S. Lee, C. Burnett, D. Cho, D. Kau, A. Fazio, and K. Soumyanath. Enabling high-performance mixed-signal system-on-a-chip (SoC) in high performance logic CMOS technology. In *Proc. IEEE VLSI Circ. Symp.*, 2002.

[4] V. Agarwal, S. Keckler, and D. Burger. The effect of technology scaling on microarchitectural structures. Tech Report TR2000-02, The University of Texas at Austin, Department of Electrical and Computer Engineering, 2000.

[5] International technology roadmap for semiconductors. http://public.itrs.net/.

[6] S. Das, A. Chandrakasan, and R. Reif. Calibration of Rent's-rule models for three-dimensional integrated circuits. *IEEE Trans. on VLSI Systems*, to appear.

[7] J. A. Burns, C. Keast, K. Warner, P. Wyatt, and D. Yost. Fabrication of 3-dimensional integrated circuits by layer transfer of fully depleted SOI circuits. In *Proc. MRS Symp. G*, volume 768, April 2003.

[8] A. Rahman and R. Reif. System-level performance evaluation of three-dimensional integrated circuits. *IEEE Trans. on VLSI Systems*, 8(6):671–678, 2000.

[9] A. Rahman, S. Das, R. Reif, and A. Chandrakasan. Wiring requirement and three-dimensional integration technology for field-programmable gate arrays. *IEEE Trans. on VLSI Systems*, 2003.

[10] C. W. Eichelberger. Three-dimensional multichip module system. United States Patent 5,111,278, May 1992.

[11] C. Val. 3-D packaging – applications of vertical multichip modules (MCM-V) for microsystems. In *Proc. IEEE/CPMT IEMT Symp.*, 1994.

[12] J. M. Stern, S. P. Larcombe, P. A. Ivey, N. L. Seed, A. J. Shelley, and N. J. Goodenough. Design and evaluation of an epoxy three-dimensional multichip module. *IEEE Trans. Comp. Pack. Man. Tech. B*, 19(1):188–194, Feb. 1996.

[13] S. P. Larcombe and P. A. Ivey. An ultra high density technology for microsystems. *Microelectronics International*, pages 15–18, Sept. 1996.

[14] E. Beyne. 3D interconnection and packaging: Impending reality or still a dream? In *Proc. ISSCC*, 2004.

[15] E. Beyne. Technologies for very high bandwidth electrical interconnects between next generation VLSI circuits. In *Proc. IEDM*, pages 23.3.1–23.3.4, 2001.

[16] G. Carchon, K. Vaesen, S. Brebels, W. De Raedt, E. Beyne, and B. Nauwelaers. Multilayer thin-film MCM-D for the integration of high-performance RF and microwave circuits. *IEEE Trans. Comp. Pack. Tech.*, 24(3):510–519, September 2001.

[17] J. H. Lau. *Low Cost Flip Chip Technologies: For DCA, WLCSP, and PBGA Assemblies.* McGraw-Hill, New York, 2000.

[18] G. Roos, B. Hoefflinger, M. Schubert, and R. Zingg. Manufacturability of 3D-epitaxial-lateral-overgrowth CMOS circuits with three stacked channels. *Microelectronic Engineering*, 15:191–194, 1991.

[19] V. Subramanian, P. Dankoski, L. Degertekin, B. T. Khuri-Yakub, and K. C. Saraswat. Controlled two-step solid-phase crystallization for high-performance polysilicon TFTs. *IEEE Electron Device Letters*, 18:378–381, Aug. 1997.

[20] M. G. Johnson, T. H. Lee, et al. Vertically stacked field programmable nonvolatile memory and method of fabrication. United States Patent 6,351,406, Nov. 2000.

[21] A. Fan, A. Rahman, and R. Reif. Copper wafer bonding. *Electrochemical and Solid-State Letters*, 2:534–536, 1999.

[22] S. Mick, J. Wilson, and P. Franzon. 4 gbps high-density AC coupled interconnection. In *Proc. CICC*, May 2002.

[23] Y. Kwon, A. Jindal, J. J. McMahon, J.-Q. Lu, R. J. Gutmann, and T. S. Cale. Dielectric glue wafer bonding for 3D ICs. In *Proc. MRS*, Spring 2003.

[24] L. Xue, C. C. Liu, H.-S. Kim, S. Kim, and S. Tiwari. Three-dimensional integration: Technology, use, and issues for mixed-signal applications. *IEEE Trans. Elect. Dev.*, 50(3):601–609, 2003.

[25] R. Reif, A. Fan, K.-N. Chen, and S. Das. Fabrication technologies for three-dimensional integrated circuits. In *Proc. ISQED*, 2002.

[26] A. Masaki and M. Yamada. Equations for estimating wire length in various types of 2-D and 3-D system packaging structures. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 10:190–198, 1987.

[27] K. C. Saraswat, K. Banerjee, A. R. Joshi, P. Kalavade, P. Kapur, and S. J. Souri. 3-D ICs: Motivation, performance analysis, and technology. In *Proc. ESSCIRC*, 2000.

[28] S. J. Souri, K. Banerjee, A. Mehrotra, and K. Saraswat. Multiple Si layer ICs: Motivation, performance analysis, and design implications. In *Proc. 37th DAC*, pages 213–220, 2000.

[29] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proceedings of the IEEE*, 89(5), May 2001.

[30] J. Joyner, P. Zarkesh-Ha, J. Davis, and J. Meindl. A three-dimensional stochastic wire-length prediction for variable separation of strata. In *Proc. IITC*, pages 123–125, 2000.

[31] S. Im and K. Banerjee. Full chip thermal analysis of planar (2-D) and vertically integrated (3-D) high performance ICs. In *Proc. IEDM*, 2000.

[32] A. Rahman, A. Fan, and R. Reif. Thermal analysis of three-dimensional (3-D) integrated circuits (ICs). In *Proc. IITC*, pages 157–159, Jun. 2001.

[33] L. McIlrath and P. M. Zavracky. An architecture for low-power real time image analysis using 3D silicon technology. In *Proc. SPIE AeroSense Symp.*, April 1998.

[34] W. Meleis, M. Leeser, P. Zavracky, and M. Vai. Architectural design of a three dimensional fpga. In *Proceedings of the 17th Conference on Advanced Research in VLSI (ARVLSI)*, pages 256–268, September 1997.

[35] A. Rahman. *System-Level Performance Evaluation of Three-Dimensional Integrated Circuits.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02139, January 2001.

[36] J. Mayega, O. Erdogan, P. M. Belemjian, K. Zhou, J. F. McDonald, and R. P. Kraft. 3D direct vertical interconnect microprocessors test vehicle. In *Proc. GLSVLSI*, pages 141–146, 2003.

[37] http://www.artisan.com.

[38] K. D. Boese, A. B. Kahng, and S. Mantik. On the relevance of wire load models. In *Proc. ACM/IEEE Intl. Conf. on SLIP*, pages 91–98, 2001.

[39] Y. Deng and W. Maly. Interconnect characteristics of 2.5-d system integration scheme. In *Proc. ISPD*, pages 171–175, April 2001.

[40] K. Kozminski. Benchmarks for layout synthesis. In *Proc. 28th DAC*, pages 265–270, 1991.

[41] C. Sechen. *VLSI Placement and Global Routing Using Simulated Annealing.* Kluwer Academic Publishers, Boston, 1988.

[42] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[43] R. Sarker and C. Newton. Solving a multiple objective linear program using simulated annealing. In *Proc. APORS*, 2000.

[44] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17:219–229, 1970.

[45] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich. GORDIAN: VLSI placement by quadratic programming and slicing optimization. *IEEE Transactions on Computer-Aided Design*, 10(3):356–365, 1991.

[46] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *Proc. 35th DAC*, 1998.

[47] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection alone produce routable placements? In *Proceedings of the 37th Design Automation Conference*, pages 477–482, 2000.

[48] A. E. Dunlop and B. W. Kernighan. A procedure for placement of standard cell VLSI circuits. In *IEEE Transactions on Computer-Aided Design*, pages 92–98, 1985.

[49] S. Dutt and W. Deng. A probability-based approach to VLSI circuit partitioning. In *Proc. 33rd DAC*, pages 100–105, 1996.

[50] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proc. 19th DAC*, pages 175–181, 1982.

[51] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 1970.

[52] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Applications in VLSI design. In *Proceedings of the 34th Design Automation Conference*, pages 526–529, 1997.

[53] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Improved algorithms for hypergraph bipartitioning. In *Proc. ASP-DAC*, pages 661–666, 2000.

[54] U. V. Catalyurek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693, 1999.

[55] P. H. Madden. Partitioning by iterative deletion. In *Proc. ISPD*, pages 83–89, April 1999.

[56] A. B. Kahng, A. E. Caldwell, and I. L. Markov. Optimal partitioners and end-case placers for standard-cell layout. In *Proc. ISPD*, pages 90–96, April 1999.

[57] N. A. Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, Boston, 1993.

[58] M. Burstein and R. Pelavin. Hierarchical wire routing. *IEEE Transactions on Computer-Aided Design*, CAD-2(4):223–234, 1983.

[59] C. Y. Lee. An algorithm for path connection and its applications. *IRE Trans. Elect. Comp.*, 1961.

[60] F. Hadlock. A shortest path algorithm for grid graphs. *Networks*, 7(4):323–334, 1977.

[61] J. Soukup. Fast maze router. In *Proc. 15th DAC*, pages 100–102, 1978.

[62] S. M. Alam, D. E. Troxel, and C. V. Thompson. A comprehensive layout methodology and layout-specific circuit analyses for three-dimensional integrated circuits. In *Proc. ISQED*, March 2002.

[63] J. Ousterhout et al. Magic: A VLSI layout system. In *Proceedings of the 21st Design Automation Conference*, pages 152–159, 1984.

[64] T. Tanprasert. An analytical 3-D placement that reserves routing space. In *Proc. ISCAS*, pages III–69–III–72, May 2000.

[65] S. T. Obenaus and T. H. Szymanski. *Placement Benchmarks for 3-D VLSI*, pages 447–455. Proc. IFIP 10th Int'l Conf. on VLSI: SoC. Kluwer, Deventer, The Netherlands, 1999.

[66] S. T. Obenaus and T. H. Szymanski. Gravity: Fast placement for 3-D VLSI. *ACM TODAES*, 8(3), July 2003.

[67] S. Das. Design and implementation of three-dimensional logic structures. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02139, May 2000.

[68] W.-H. Huang and Y. V. Lei. 2-D and 3-D performance analysis of a digitally controlled CMOS class-E amplifier. Technical report, Massachusetts Institute of Technology, Course 6.374, Fall 2002.

[69] E. Basha, K. Butler, and P. Griffin. 3D analog-to-digital converter. Technical report, Massachusetts Institute of Technology, Course 6.374, Fall 2003.

[70] A. L. Rosenberg. *Three-dimensional integrated circuitry*, pages 69–80. VLSI Systems and Computations. Computer Science Press, Rockville, MD, 1981.

[71] A. L. Rosenberg. Three-dimensional VLSI: A case study. *Journal of the ACM*, 30(3):397–416, 1983.

[72] F. T. Leighton and A. L. Rosenberg. Three-dimensional circuit layouts. *SIAM Journal on Computing*, 15(3):793–813, 1986.

[73] C. E. Leiserson. VLSI theory and parallel supercomputing. MIT/LCS/TM 402, Massachusetts Institute of Technology, Laboratory for Computer Science, May 1989.

[74] D. Stroobandt and V. Campenhout. Estimating interconnection length in three-dimensional computer systems. *IEICE Trans. on Information and Systems*, 80(10):1024–1031, 1997.

[75] A. Rahman, A. Fan, J. Chung, and R. Reif. Wire-length distribution of three-dimensional integrated circuits. In *Proc. IITC*, pages 233–235, 1999.

[76] B. S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Transactions on Computers*, C-20(12), 1971.

[77] P. Christie and D. Stroobandt. The interpretation and application of Rent's rule. *IEEE Trans. on VLSI Systems*, 8(6):639–648, December 2000.

[78] J. Davis, V. K. De, and J. D. Meindl. A stochastic wire-length distribution for gigascale integration (GSI) – Part I: Derivation and validation. *IEEE Transactions on Electron Devices*, 45(3):580–589, 1998.

[79] X. Yang, E. Bozorgzadeh, and M. Sarrafzadeh. Wirelength estimation based on Rent exponents of partitioning and placement. In *Proc. ACM/IEEE Intl. Conf. on SLIP*, 2001.

[80] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Fast standard-cell placement for large circuits. In *IEEE International Conference on Computer-Aided Design*, pages 260–263, 2000.

[81] X. Yang, B.-K. Choi, and M. Sarrafzadeh. Routability driven white space allocation for fixed-die standard-cell placement. In *Proc. ISPD*, April 2002.

[82] C. J. Alpert. The ISPD98 circuit benchmark suite. In *Proc. ISPD*, pages 80–85, April 1998.

[83] A. Srivastava, C. Chen, and M. Sarrafzadeh. Timing-driven gate duplication in the technology-independent stage. *IEICE Trans. Fund. E.C.C.S.*, E84-A:2673–2680, Nov. 2001.

[84] E. Lehman, Y. Watanabe, J. Grodstein, and H. Harkness. Logic decomposition during technology mapping. In *Proc. ICCAD*, 1995.

[85] S.-L. Ou and M. Pedram. Timing-driven placement based on partitioning with dynamic cut-net control. In *Proc. 37th DAC*, pages 472–476, 2000.

[86] B. Halpin, C. Chen, and N. Sehgal. Timing driven placement using physical net constraints. In *Proc. 38th DAC*, pages 780–783, 2001.

[87] M. Pedram. Power minimization in IC design: Principles and applications. *ACM Trans. on DAES*, 1:3–56, 1996.

[88] S. Devadas and S. Malik. A survey of optimization techniques targeting low-power VLSI circuits. In *Proc. 32nd DAC*, pages 242–247, 1995.

[89] R. Zhang, K. Roy, C.-K. Koh, and D. B. Janes. Stochastic interconnect modeling, power trends, and performance characterization of 3-d circuits. *IEEE Trans. Elect. Dev.*, 48(4), April 2001.

[90] R. Zhang, K. Roy, C.-K. Koh, and D. B. Janes. Power trend and performance characterization of 3-dimensional integration for future technology generations. In *Proc. ISQED*, pages 217–222, March 2001.

[91] W. E. Donath, R. J. Norman, B. K. Agrawal, S. E. Bello, S. Y. Han, J. M. Kurtzberg, P. Lowy, and R. I. McMillan. Timing driven placement using complete path delays. In *Proc. 27th DAC*, pages 84–89, 1990.

[92] C. Ababei, N. Selvakkumaran, K. Bazargan, and G. Karypis. Multi-objective circuit partitioning for cutsize and path-based delay minimization. In *Proc. ICCAD*, 2002.

[93] Y.-C. Chou and Y.-L. Lin. A performance-driven standard-cell placer based on a modified force-directed algorithm. In *Proc. ISPD*, pages 24–29, 2001.

[94] R. Zhang, K. Roy, C.-K. Koh, and D. B. Janes. Stochastic wire-length and delay distributions of 3-dimensional circuits. In *Proc. ICCAD*, pages 208–213, 2000.

[95] A. Wang. *An Ultra-Low Voltage FFT Processor Using Energy-Aware Techniques.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02139, December 2003.

[96] http://www.opencores.org.

[97] H. Thon. Three red-hot boxed cooler alternatives for the Athlon XP3200+. http://www.tomshardware.com/cpu/ 20030917/boxed_cooler-04.html, September 2003.

[98] A. A. Keshavarz, P. Khare, and R. K. Sampson. Comprehensive modeling of mos transistors in a 0.35um technology for analog and digital applications. In *Proc. Intl. Conf. MSM*, 2002.

[99] W. Liao, F. Lei, and L. He. Microarchitecture level power and thermal simulation considering temperature dependent leakage model. In *Proc. ISLPED*, pages 211–216, 2003.

[100] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design: A Systems Perspective.* Addison-Wesley, Reading, MA, 2nd edition, 1994.

[101] R. Blish and N. Durrant. Semiconductor device reliability failure models. Technology Transfer 00053955A-XFR, International SEMATECH, May 2000.

[102] K. Banerjee, M. Pedram, and A. H. Ajami. Analysis and optimization of thermal issues in high-performance VLSI. In *Proc. ISPD*, 2001.

[103] K. Banerjee, A. Mehrotra, A. Sangiovanni-Vincentelli, and C. Hu. On thermal effects in deep sub-micron vlsi interconnects. In *Proc. 36th DAC*, pages 885–891, 1999.

[104] J. R. Black. Electromigration – a brief survey and some recent results. *IEEE Trans. Electron Devices*, 16:338–347, 1969.

[105] K. Banerjee, A. Amerasekera, N. Cheung, and C. Hu. High-current failure model for VLSI interconnects under short-pulse stress conditions. *IEEE Elect. Dev. Lett.*, 18(9):405–407, 1997.

[106] A. H. Ajami, M. Pedram, and K. Banerjee. Effects of non-uniform substrate temperature on the clock signal integrity in high performance designs. In *Proc. CICC*, 2001.

[107] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas. Impact of interconnect variations on the clock skew of a gigahertz microprocessor. In *Proc. 37th DAC*, June 2000.

[108] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon. High-performance microprocessor design. *IEEE JSSC*, 33(5), May 1998.

[109] C.-H. Tsai and S. Kang. Standard cell placement for even on-chip thermal distribution. In *Proc. ISPD*, pages 179–184, 1999.

[110] K.-K. Lee, E. J. Paradise, and S. K. Lim. Thermal-driven circuit partitioning and floorplanning with power optimization. Technical Report GIT-CERCS-03-07, The Georgia Institute of Technology Center for Experimental Research in Computer Systems, 2003.

[111] G. Chen and S. Sapatnekar. Partition-driven standard cell thermal placement. In *Proc. ISPD*, pages 75–80, 2003.

[112] B. Goplen and S. Sapatnekar. Efficient thermal placement of standard cells in 3D ICs using a force directed approach. In *Proc. ICCAD*, 2003.

[113] D. B. Tuckerman and R. F. W. Pease. High-performance heat sinking for VLSI. *IEEE Electron Device Letters*, 2(5):126–129, May 1981.

[114] D. K. Das and R. K. Prabhudesai. *Chemical Engineering License Review.* Engineering Press, San Jose, 1996.

[115] Y. A. Çengel. *Heat Transfer: A Practical Approach.* McGraw-Hill, Boston, 1998.

[116] F. Kreith. *Principles of Heat Transfer.* Harper and Row, New York, 3rd edition, 1973.

[117] L. Zhang, J.-M. Koo, L. Jiang, M. Asheghi, K. E. Goodson, J. G. Santiago, and T. W. Kenny. Measurements and modeling of two-phase flow in microchannels with nearly constant heat flux boundary conditions. *Journal of MEMS*, 11(1):12–19, February 2002.

[118] V. K. Samalam. Convective heat transfer in microchannels. *J. Electron. Mater.*, 18(5):611–618, September 1989.

[119] R. W. Knight, D. J. Hall, J. S. Goodling, and R. C. Jaeger. Heat sink optimization with application to microchannels. *IEEE Trans. Compon., Hybr., Manufact. Technol.*, 15:832–842, October 1992.

[120] Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM Press, Philadelphia, 2nd edition, 2003.

[121] C. Hu. Future CMOS-scaling and reliability. In *Proc. IEEE*, volume 81, pages 682–689, 1993.

[122] B. Davari, R. H. Dennard, and G. G. Shahidi. CMOS-scaling for high performance and low power – the next ten years. In *Proc. IEEE*, volume 83, pages 595–606, 1995.

[123] M. Horowitz, R. Ho, and K. Mai. The future of wires. In *SRC Workshop on Interconnects for SOCs*, May 1999.

[124] D. Sylvester and K. Keutzer. Rethinking deep-submicron circuit design. *IEEE Computer*, 32(11):25–33, November 1999.

[125] T. Sakurai and A. R. Newton. Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas. *IEEE JSSC*, 25(2):584–594, 1990.

[126] Processors and performance: When do GHz hurt? Panel discussion, ISSCC 2004.

[127] R. Kumar. Interconnect and noise immunity design for the Pentium® 4 processor. *Intel Technology Journal*, 1st qtr. 2001.

[128] D. Leenaerts and P. de Vreede. Influence of substrate noise on RF performance. In *Proc. ESSCIRC*, 2000.

[129] M. van Heijningen, J. Compiet, P. Wanbacq, S. Donnay, M. G. E. Engels, and I. Bolsens. Analysis and experimental verification of digital substrate noise generation for epi-type substrates. *IEEE JSSC*, 35(7):1002–1008, July 2000.

[130] M. van Heijningen, M. Badaroglu, S. Donnay, H. De Man, G. Gielen, M. Engels, and I. Bolsens. Substrate noise generation in complex digital systems: Efficient modeling and simulation methodology and experimental verification. In *ISSCC Digest of Technical Papers*, pages 342–343, 2001.

[131] http://www.cadence.com/products/ digital_ic/pks/index.aspx.

[132] http://www.eda.org/vhdl-ams/.

[133] http://www.eda.org/verilog-ams/.

[134] http://openeda.org/download_lefdef.html.

[135] http://vlsicad.cs.ucla.edu/ GSRC/ bookshelf/ Slots/ Placement/ plFormats.html.