Integrated Circuits and Systems

Series Editor
Anantha Chandrakasan, Massachusetts Institute of Technology
Cambridge, Massachusetts

For other titles published in this series, go to
www.springer.com/series/7236

Ron Ho • Robert Drost
Editors

Coupled Data Communication Techniques for High-Performance and Low-Power Computing

Editors
Ron Ho
Oracle Corporation
Sun Labs
VLSI Research Group
16 Network Circle
UMPK 16-161
Menlo Park, CA 94025
USA
ron.ho@oracle.com

Robert Drost
Los Altos, CA 94024
USA

ISSN 1558-9412
ISBN 978-1-4419-6587-5 e-ISBN 978-1-4419-6588-2
DOI 10.1007/978-1-4419-6588-2
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010927932

© Springer Science+Business Media, LLC 2010
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

For Christina, Sawyer, and Finley – RH
For Sharon and Juliet – RJD

Foreword

Wafer-scale integration has long been the dream of system designers. Instead of chopping a wafer into a few hundred or a few thousand chips, one would just connect the circuits on the entire wafer. What an enormous capability wafer-scale integration would offer: all those millions of circuits connected by high-speed on-chip wires. Unfortunately, the best known optical systems can provide suitably fine resolution only over an area much smaller than a whole wafer. There is no known way to pattern a whole wafer with transistors and wires small enough for modern circuits.

Statistical defects present a firmer barrier to wafer-scale integration. Flaws appear regularly in integrated circuits; the larger the circuit area, the more probable there is a flaw. If such flaws were the result only of dust one might reduce their numbers, but flaws are also the inevitable result of small scale. Each feature on a modern integrated circuit is carved out by only a small number of photons in the lithographic process. Each transistor gets its electrical properties from only a small number of impurity atoms in its tiny area. Inevitably, the quantized nature of light and the atomic nature of matter produce statistical variations in both the number of photons defining each tiny shape and the number of atoms providing the electrical behavior of tiny transistors. No known way exists to eliminate such statistical variation, nor may any be possible.

Proximity communication, or coupled data communication in general, may make possible the long-sought dream of wafer scale integration. Proximity communication permits assembly of wafer-scale systems from small parts. We can make circuit chips small enough for low defect rates, cast aside bad chips, and reassemble the good chips into wafer-scale systems.

Two properties of proximity communication suit it to wafer-scale use. First, quality: the connections between chips are nearly as good as wires on a single chip. As this book describes, proximity connections are fast, occupy small area, and consume little energy. Second, and I think much more important, is replacement: proximity communication permits one to replace chips in a big system. Together, quality and replacement make wafer-scale integration possible. Because I think replacement is so important, Im going to devote a few more lines to it.

What makes replacement possible? Proximity communication needs neither welds nor solder. The parts are joined electrically only by the electric fields between them. These fields pass right through the top layers of glass that protect the chips. Within error limits, the communication is also insensitive to chip separation and chip alignment. If one chip in a wafer-scale assembly of hundreds of chips proves unsuitable because of a hidden defect, or through aging, or simply for product upgrade, no physical bonds prevent its replacement.

I believe that replacement will prove most useful for test. A complete system could serve as a jig that would test fresh chips in their real environment. Each fresh chip would spend only long enough in the complete system for a thorough test. A test jig smaller than a complete system might also serve to test only a single type of chip, providing it an environment indistinguishable from a full system. Such a test jig would have full speed access to every connection to or from the fresh chip. I see a huge potential for replacement to simplify and improve test.

I also see that replacement may permit a profound change in the business alliances that produce products. Without the ability to replace, one bad chip destroys an entire multi-chip module, making specialization in module assembly a poor business. Because one bad chip spoils the entire module, a contractor who assembles multi-chip modules must take responsibility not only for defects in his own process, but also for defects in separate chips. This dual responsibility is a very high barrier to contract assembly.

Board-level assembly houses are common because they avoid this dual barrier in two ways. First, not only is board-level assembly an old art with a well known low defect density but also it uses packaged and well tested parts. Second and more important, at the board-level some, albeit limited, replacement is possible. It is possible to remove and reuse at least the high-value chips on a board-level assembly, greatly reducing the high cost of bad parts. I believe that because proximity communication permits replacement it will also foster wafer-scale assembly houses.

Bob Johnson, formerly technical head of Burroughs, talked about using conductive grease to connect the ordinary pads on chips placed face-to-face. A large area of thin grease between facing pads would provide a connection. The thinner and much longer layer of grease reaching to other pads would produce small but manageable cross talk. I merely replaced Johnson's grease with electric fields. Robert Drost's fiendishly clever diagonal arrangement of pads greatly reduces cross talk.

Bob Bosnyak designed and measured some early proximity communication test chips. I recall one flawed ring oscillator test chip built for us by the MOSIS foundry service. The flaw turned out to be total omission of the metal plates on adjacent levels of metal that were to form the bulk of Bosnyak's test structure. Nevertheless, the test chip worked, albeit at a mystifying small fraction of its intended speed. The mystery vanished when we discovered the omitted plates. MOSIS rebuilt the test chip for free.

The late Bob Proebsting, a pioneer and life-long designer of fine memory parts, contributed to us much knowledge about sense amplifiers. For a period, the authors of this book were, in effect, Proebsting's post-doc students. As usual in such relationships, both the brilliant teacher and the apt students took much delight from the

process. It was my joy to assemble such a mass of brainpower and to watch both its progress and the continuing delight of its participants.

Portland, Oregon, September 2009 *Ivan Sutherland*

Contents

Part I Introduction

1 Introduction to Coupled Data Technologies 3
Ron Ho, Robert Drost
1.1 Life has been good 3
1.2 Faster computers tomorrow 4
1.2.1 The end of Moore's Law 7
1.2.2 The arguments against–and for–multiple chips 7
1.3 Coupled data communication 8
1.3.1 This book 9
References 10

Part II Overview of 3D Technologies

2 Power delivery, signaling and cooling for 2D and 3D integrated systems 13
Muhannad Bakir, Gang Huang and Bing Dang
2.1 Introduction 13
2.2 Evolution of conventional silicon ancillary technologies: A brief overview 14
2.3 Novel silicon ancillary technologies 18
2.3.1 Optical I/Os 23
2.3.2 Fluidic I/Os for single and 3D chips 26
2.4 Power delivery for 2D and 3D systems 31
2.4.1 Power delivery and design implications of 2D systems 34
2.4.2 Power delivery and design implications of 3D systems 38
2.5 Conclusion 43
References 45

Part III Coupled Data Technologies

3 Capacitive Coupled Communication 51
David Hopkins, Alex Chow, Frankie Liu, Dinesh D. Patil, Hans Eberle
3.1 Introduction 51
3.2 An electrical model of capacitive interchip communication 53
3.2.1 Crosstalk mitigation 56
3.2.2 Simulation results 56
3.3 Transmitting data 61
3.4 Receiving data 62
3.4.1 Attenuation 62
3.4.2 Loss of DC information 63
3.4.3 Comparators 65
3.4.4 Receiver sizing 66
3.4.5 Timing schemes 67
3.5 Two-dimensional arrays 68
3.6 Measurement results 70
3.6.1 Voltage waterfall 70
3.6.2 Timing waterfall 71
3.6.3 Combined eye diagram 72
3.6.4 BER versus chip separation 72
3.7 Prototype application: a high-radix switch 73
References 77

4 Inductive Coupled Communications 79
Noriyuki Miura, Takayasu Sakurai, and Tadahiro Kuroda
4.1 Introduction 79
4.2 Inductive-coupling channel 80
4.2.1 Overview of channel characteristics 80
4.2.2 Range extendability 83
4.2.3 Coupling strength through Si substrate 84
4.2.4 Crosstalk 85
4.3 Inductive-coupling transceiver 86
4.3.1 Signaling 87
4.3.2 Coil design 89
4.3.3 Transceiver circuit design 91
4.3.4 Inter-chip communications 92
4.4 Power reduction techniques 93
4.4.1 Pulse shaping 94
4.4.2 Daisy chain transmitter 98
4.5 High-speed techniques 100
4.5.1 Asynchronous transceiver 101
4.5.2 Burst transmission 104
4.6 Crosstalk reduction techniques 106
4.6.1 Time interleaving 107

4.6.2 Differential coil 109
4.7 Application I: memory stacking 111
4.7.1 Homogenous chip stacking 114
4.7.2 Inductive-coupling up/down repeater 114
4.7.3 Test chip measurement 117
4.8 Application II: processor and memory stacking 118
4.8.1 Heterogenous chip stacking 119
4.8.2 Interface design 120
4.8.3 Test chip measurement 121
4.9 Conclusion 122
References 124

5 Use of AC Coupled Interconnect in Contactless Packaging 127
Paul Franzon
5.1 Introduction: Why use ACCI? 127
5.1.1 Chapter outline 129
5.2 Historical Perspectives 129
5.3 Capacitively Coupled Chip I/O 129
5.3.1 Capacitively Coupled Channel Design 130
5.3.2 ACCI Circuits 137
5.3.3 ACCI Packaging 141
5.4 Mid-channel Capacitively Coupled Structures 142
5.5 Inductively Coupled Connectors and Sockets 146
5.6 Conclusions and Future Perspectives 151
References 152

Part IV Enabling Coupled Data Technologies

6 Aligning chips face-to-face for dense capacitive communication 157
John E. Cunningham, Ashok V. Krishnamoorthy, Ivan Shubin, James G. Mitchell, Xuezhe Zheng
6.1 Introduction 157
6.2 Aligning chips face-to-face 158
6.2.1 Power and ground connections between coupled chips . . . 163
6.3 A low-cost package for capacitive proximity communication 168
6.4 Array packages using bridge chips 171
References 174

Part V Extending Data Coupling Technologies

7 Delivering On-chip Bandwidth Off-chip and Out-of-box with Proximity and Optical Communication 179
Ashok V. Krishnamoorthy, Jon Lexau, Xuezhe Zheng, John E. Cunningham
7.1 Introduction 179
7.2 Photonics as a long-reach interconnect 180

7.3 Photonics on VLSI (optoelectronic VLSI) . . . 182
7.4 Proximity and photonic communication . . . 184
7.5 Test chip results . . . 185
7.6 Conclusion . . . 190
References . . . 191

8 AC Coupled Wireless Power Delivery . . . 193
Makoto Takamiya, Kohei Onizuka, and Takayasu Sakurai
8.1 Three dimensional stacked inter-chip wireless power delivery . . . 193
8.2 Prototype of wireless power transmission circuits . . . 195
8.3 Theoretical analysis and circuit improvements . . . 198
8.4 Summary . . . 203
References . . . 204

Index . . . 205

List of Contributors

Dr. Ron Ho
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: ron.ho@sun.com

Dr. Robert Drost
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: robert.drost@sun.com

Dr. Muhannad Bakir
Microelectronics Research Center, Georgia Institute of Technology, 791 Atlantic Dr. NW, Atlanta, GA 30332-0269, USA, e-mail: muhannad.bakir@mirc.gatech.edu

Alex Chow
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: alex.chow@sun.com

Dr. John E. Cunningham
Sun Microsystems Chief Technology Organization, 9515 Towne Centre Drive, San Diego, CA 92121, USA, e-mail: john.cunningham@sun.com

Dr. Bing Dang
IBM T. J. Watson Research Center, 1101 Kitchawan Rd, RM 6-242, Yorktown Heights, NY 10598, USA, e-mail: dangbing@us.ibm.com

Dr. Hans Eberle
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: hans.eberle@sun.com

Prof. Paul Franzon
Department of Electrical and Computer Engineering, North Carolina State University, Box 7914, Raleigh, NC, 27695, USA, e-mail: paulf@ncsu.edu

David Hopkins
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025,

USA, e-mail: robert.hopkins@sun.com

Dr. Gang Huang
Intel Corporation, Ultra Mobility Group, 1501 S. MO-Pac Expy, Austin, TX 78746 USA, e-mail: gang.huang@intel.com

Dr. Ashok V. Krishnamoorthy
Sun Microsystems Chief Technology Organization, 9515 Towne Centre Drive, San Diego, CA 92121, USA, e-mail: ashok.krishnamoorthy@sun.com

Professor Tadahiro Kuroda
Department of Electrical Engineering, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223-8522, JAPAN, e-mail: kuroda@elec.keio.ac.jp

Jon K. Lexau
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: jon.lexau@sun.com

Dr. Frankie Liu
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: frankie.liu@sun.com

Professor Noriyuki Miura
Department of Electrical Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522 JAPAN, e-mail: miura@kuro.elec.keio.ac.jp

Dr. James G. Mitchell
Sun Microsystems Chief Technology Organization, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: jim.mitchell@sun.com

Dr. Kohei Onizuka
Formerly with the Institute of Industrial Science, University of Tokyo, and now with Toshiba Corporation.

Dr. Dinesh D. Patil
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: dinesh.d.patil@sun.com

Professor Takayasu Sakurai
Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, JAPAN, e-mail: tsakurai@iis.u-tokyo.ac.jp

Dr. Ivan Shubin
Sun Microsystems Chief Technology Organization, 9515 Towne Centre Drive, San Diego, CA 92121, USA, e-mail: ivan.shubin@sun.com

Professor Makoto Takamiya
VLSI Design and Education Center, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, JAPAN, e-mail: mtaka@iis.u-tokyo.ac.jp

Dr. Xuezhe Zheng
Sun Microsystems Chief Technology Organization, 9515 Towne Centre Drive, San Diego, CA 92121, USA, e-mail: xuezhe.zheng@sun.com

Part I
Introduction

Chapter 1
Introduction to Coupled Data Technologies

Ron Ho, Robert Drost

1.1 Life has been good

The past quarter-century has seen an explosive growth in the performance of computer systems. One of the first widely popular personal computers was a mid-1980s IBM PC, running on a 4.77 MHz Intel 8088 processor, stuffed with 256 KB of system memory (plus another 384 KB on an expansion card), displaying 640x200 black-and-white graphics, and storing data on 360 KB 5.25-inch floppy disks. In 2009, a typical workstation configuration sold by Sun Microsystems, the Ultra 24 Workstation, used a 3 GHz Intel Quad Core 2 processor with 8 GB of memory, displayed 2560x1600 graphics on a 30-inch LCD monitor using an Nvidia Quadro NVS 290 accelerator card, with up to 1.8 TB of Serial-Attached SCSI drives spinning at 15 krpm.

Both systems cost around $4000 in contemporary dollars.

The enormous advancement in price-performance between these computer systems came from improvements in many different technologies, including storage media, displays, software systems, and so on. But certainly a large part of it was because VLSI semiconductors, and high-end microprocessors and memories in particular, have gotten faster.

Figure 1.1 shows the historical performance of microprocessors, normalized to the SpecINT2000 benchmark [1], and how it has seen a remarkable 35% cumulative annual growth rate over the past twenty-five years – a growth curve seen by virtually no other industry. The natural question prompted by this chart is, "can this growth curve continue?" Or, for the readers of this book, "what must designers do to enable it to continue?"

This growth in performance is popularly, though somewhat incorrectly, fully attributed to "Moore's Law." This is what Carver Mead at CalTech called Gordon

Dr. Ron Ho and Dr. Robert Drost
Sun Microsystems Research Labs, 16 Network Circle, Menlo Park, CA 94025, USA, e-mail: {ron.ho},{robert.drost}@sun.com

R. Ho and R. Drost (eds.), *Coupled Data Communication Techniques for High-Performance and Low-Power Computing*, Integrated Circuits and Systems, DOI 10.1007/978-1-4419-6588-2_1,
© Springer Science+Business Media, LLC 2010

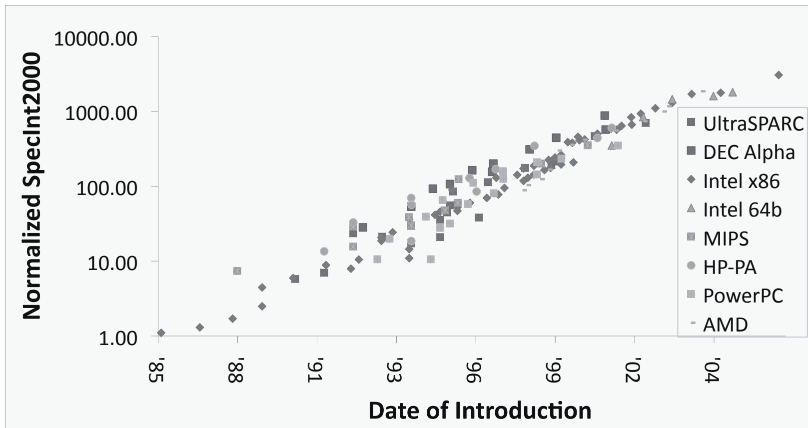

Fig. 1.1 Processor performance scaling over the past twenty-five years [3].

Moore's now-famous 1965 extrapolation of transistor density scaling [2]. Moore had argued that when optimized for lowest total cost, integrated circuit chips would, over time, contain an ever-growing number of transistors. Too few transistors per chip, and the fixed overhead of manufacturing and packaging the chips would dominate their cost; too many transistors, and random defects would excessively reduce the yield of good chips and hence increase their cost. But the *right* number of transistors–the number that minimized cost–would continue to grow, as wafer sizes increase and transistor dimensions decrease.

In reality, transistor density scaling has only partly fueled the growth in computer systems performance. Equally important have been rapid advances in raw transistor speeds and in aggressive design techniques, as we discuss next.

1.2 Faster computers tomorrow

For a new computer system to out-perform an old computer system on the same software program, it must demonstrate improvements in the product of three terms: seconds per logic gate, logic gates per clock cycle, and clock cycles per instruction [4]. The product of these three gives program execution rate, in seconds per instruction.

The number of seconds per logic gate (approximately 10^{-11} seconds, or 10 ps, in a modern 40 nm process technology) has been scaling down roughly linearly with technology for many years: a technology with half the drawn transistor dimensions as another could be expected to be twice as fast. Each new generation reduces dimensions by 70%, so this doubling of speed arrives two generations, or every five to six years.

While this improvement trend has held steady for several technology generations, designers expect it to slow down soon. This is because maintaining transistor performance directly conflicts with reducing transistor power, and power has become a primary design constraint in today's systems. As a result, transistor designers will likely choose to reduce what they have long jokingly called their "technology entitlement," and live with devices that are only slightly faster each process generation.

But even if the delay of logic gates does not reduce as many designers expect, it provides at best a 2x improvement every five to six years, or approximately a 13% annual growth rate. More must be done to match the 35% historical growth rate in computer performance.

The number of logic gates per clock cycle, when combined with the seconds per logic gate, gives clock frequency, which is 2–5 GHz in modern processors. Logic gates per clock cycle directly measure the aggressiveness of the processor design: a CPU with thirty gates per clock cycle is much less aggressive than one with only ten gates per clock cycle; its designers have much more time per cycle to perform computation or communication. What is the limit to this design aggressiveness? Over

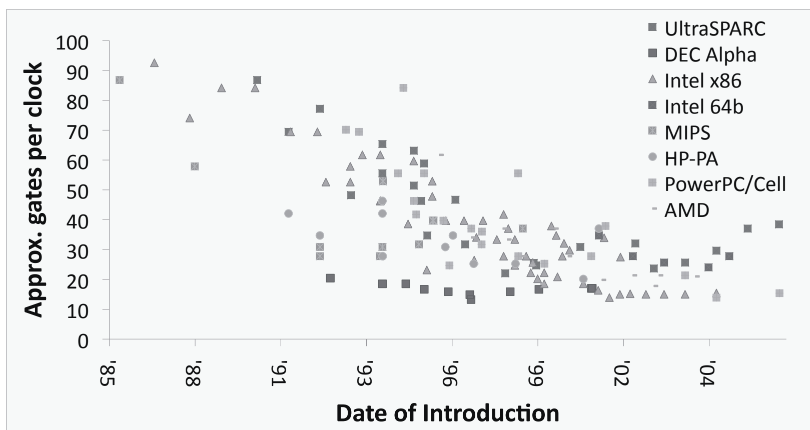

Fig. 1.2 Scaling of logic gates per clock as a function of technology generation.

the past twenty years, the number of logic gates per cycle has fallen as the aggressiveness of designs has increased. Pre-Pentium processors used around 100 gates per cycle. Today, the industry has settled in the range of 15–30 gates per cycle. Collectively we now understand that achieving the lower end of that range is possible but disproportionately expensive: building so-called "short-tick" machines requires much more effort and care in clock distribution, parasitic extraction, timing verification, and min-path methodology. For example, a modern processor has a clocking overhead of nearly two gates per cycle, so a ten gate-per-clock design would thus have only eight gate delays in which to do work, barely enough time to complete a 64-b integer addition. While doable, such designs consume not only extra design

resources and non-recurring engineering (NRE) costs but also significantly extra power in the design.

Therefore, the number of logic gates per cycle will most likely not fall any further. Combined with the argument above for seconds per logic gate, this predicts slow changes in clock frequency, on the order of 13% per year and likely even lower.

Therefore, the way to continue to improve processor performance must come from reducing the number of cycles per instruction. This arises through increased parallelism: pipelined or superscalar execution, vector processing, and speculation all aim to increase the number of operations concurrently executed[1]. At a larger scale, processors with multiple cores and a shared memory can be used to divide a complex problem into separate threads. Historically, such techniques have provided the balance of the performance gains shown in Figure 1.1, with designers increasingly leveraging and targeting parallelism.

However, increased parallelism–and reduced cycles per instruction–has a cost: processors by necessity also grow increasingly complex. Larger instruction windows to winnow out code independencies require larger queues and communication structures. Multiple execution pipes require more area for more adders, multipliers, and registers, as well as the switches to access these added functional blocks. Processors packed with multiple cores need to fit not only those cores on the die, but also correspondingly more cache to keep them all fed.

This last point bears repeating: suppose we increase the number of cores on a chip. If we keep the memory-to-core ratio constant, then each core still has the same amount of cache available to it, and therefore has a consistent cache miss rate. However, because of the growing number of cores, the total aggregated miss rate for the chip will go up and put pressure on the fixed off-chip I/O bandwidth; as a result, when increasing the number of cores on a chip we must in fact disproportionately increase the cache size as well, to lower miss rates and to continue to fit inside the available total chip I/O.

Thus far the transistor density scaling provided by Moore's Law has kept up with the need for ever-complex architectures and systems, and allowed us to continue to find and to exploit parallelism on a chip. In other words, the improvements in clock cycles per instruction provided by Moore's Law scaling have combined with the historical improvements in seconds per logic gate and logic gates per clock to give the trends in Figure 1.1.

[1] In this discussion we gloss over important distinctions between instruction-level parallelism and task-level parallelism. While they are remarkably different at an architectural level, at a physical level both require similar increases in integration and hence increased transistor counts in a package.

1.2.1 The end of Moore's Law

"Is Moore's Law ending?" is a perennially-asked question in industry journals and conferences. For several very good reasons involving seemingly fundamental physics, feature size scaling has "always" been on its last legs. Yet the industry has stubbornly insisted on solving these problems and continually shrinking transistors and wires.

Today, foundries pattern structures with dimensions of a few 10s of nm using light with a wavelength of 193 nm. By rights, this ought to be impossible. Yet it is done, by using optical proximity correction, phase-shifting masks, off-axis illumination, spacer masks, and some extremely expensive diffractive lenses. Atomic thinness limits in oxide gate insulators are overcome by employing metallic gates and high-permittivity liners, which happily also help reduce gate leakage currents. And a combination of mostly-air dielectrics that reduce wires parasitics, and thick deposited metals that reduce wire resistance, have helped to keep wires from overly constraining chip performance.

Will these improvement trends continue in the next ten to twenty years? While the answer "no" has been proven wrong time and time again, recent economic limitations have now supplanted technology as the likely true limit for Moore's Law. Especially given the financial realities of the current global economic crisis, the semiconductor industry can no longer continue to enjoy a fully elastic market that supports ever-increasing global financial investment. Worse yet, new fabrication plants will each cost over 1% of the total semiconductor market, thus limiting the number of new technologies able to come on-line each year.

Gordon Moore himself pointed out that his "law" will eventually end, although he was hopeful that new technologies would delay that date–and from his talk in 2003 to the present, they certainly have. However, any industry that constantly relies on exponential growth to continue will eventually be disappointed.

Thus, Moore's Law of transistor scaling has historically combined with logic gate scaling and clock rate scaling to enable faster and faster computers. Looking forward, Moore's Law is the only scaling trend left, as gate scaling and clock rate scaling are both slowing down for design and integration reasons, and even Moore's Law will not survive through the next few technology generations.

What is a designer of high-end computer systems to do?

1.2.2 The arguments against–and for–multiple chips

Designers can achieve more complex systems either by exploiting Moore's Law scaling for a single chip or by aggregating the functionality across multiple chips. An example of the former is a recent Xeon microprocessor from Intel that occupies nearly 7 cm^2 in area and contains as many as eight full processors and a proportionally large cache [5]. An example of the latter would be an IBM Power processor with five chips integrated on a multi-chip module (MCM).