Innovative Technology for Computer Professionals

Computer

March 2004

The Zen of Overclocking

Embedded Computing Paradigms

Languages and the Computing Profession

IEEE

IEEE COMPUTER SOCIETY

http://www.computer.org

HiPC 2004

December 19 - 22, 2004
Bangalore, India

The 11th Annual International Conference on High Performance Computing (HiPC 2004) will be held in Bangalore, the Silicon Valley of India, during December 19-22, 2004. It will serve as a forum to present current work by researchers from around the world as well as highlight activities in Asia in the high performance computing area. HiPC has a history of attracting participation from reputed researchers from all over the world. HiPC 2003 was held in Hyderabad, India, and included 48 contributed papers selected from 164 submissions spanning 11 countries. HiPC 2004 will emphasize the design and analysis of high performance computing and networking systems and their scientific, engineering, and commercial applications. In addition to technical sessions of contributed paper presentations, the conference will offer invited presentations, a poster/presentation session, tutorials, and vendor presentations.

COSPONSORED BY:

Indian Institute of Information Technology, Bangalore
IEEE Computer Society Technical Committee on Parallel Processing
ACM SIGARCH
European Association for Theoretical Computer Science
IFIP Working Group on Concurrent Systems
National Association of Software and Services Companies (NASSCOM)
Manufacturers Association for Intermation Technology (MAIT)

CALL FOR PAPERS

Authors are invited to submit original unpublished manuscripts that demonstrate current research in all areas of high performance computing including design and analysis of parallel and distributed systems, embedded systems, and their applications in scientific, engineering, and commercial areas. All submissions will be peer reviewed. The paper submission deadline is May 03, 2004. Awards, sponsored by Infosys, will be awarded to outstanding contributed papers in two areas: a) Algorithms and Applications and b) Systems.

GENERAL CO-CHAIRS

Viktor K. Prasanna, University of Southern California
Uday Shukla, IBM India

VICE GENERAL CHAIR

David A. Bader, University of New Mexico

PROGRAM CHAIR

Luc Bouge, IRISA/ENS Cachan, France

PROGRAM VICE CHAIRS

Algorithms
Frédéric Desprez, INRIA Rhône-Alpes, France

Applications
Frank Mueller, North Carolina State University

Architecture
Per Stenström, Chalmers University of Technology, Sweden

Communication Networks
Ramesh Govindan, University of Southern California

Systems Software
Thilo Kielmann, Vrije Universiteit, The Netherlands

HiPC

For further information visit the HiPC Web site at
www.hipc.org

Call for Papers and Professional Engagement

27th International Conference on Software Engineering

St Louis, Missouri, USA, 15-21 May 2005
http://www.icse-conferences.org/2005/

Sponsored by ACM SIGSOFT and IEEE Computer Society-TCSE

Conference Organization

General Chair
Gruia-Catalin Roman *Washington U., St. Louis, USA*

Program Chairs
William Griswold *UC San Diego, USA*
Bashar Nuseibeh *The Open University, UK*

Experience Reports
Constance Heitmeyer *Naval Research Lab, USA*
Klaus Pohl *U. of Duisburg-Essen, Germany*

Educational Reports
Paola Inverardi *U. of L'Aquila,Italy*
Mehdi Jazayeri *TU. of Vienna, Austria*

State of the Art
David Garlan *Carnegie Mellon U., USA*
Jeff Kramer *Imperial College, UK*

State of the Practice
Wolfgang Emmerich *University College London, UK*
David Rosenblum *University College London, UK*

Extending the Discipline
John Mylopoulos *U. of Toronto, Canada*
Ian Sommerville *Lancaster U., UK*

Panels
Jeffrey Magee *Imperial College, UK*
Kevin Sullivan *U. of Virginia, USA*

Workshops & Co-located Events
Andre van der Hoek *UC Irvine, USA*
Gian Pietro Picco *Politecnico di Milano, Italy*

Tutorials
Steve Easterbrook *U. of Toronto, Canada*
Martin Glinz *U. of Zurich, Switzerland*

Demos and Posters
Prem Devanbu *UC Davis, USA*
Cecilia Mascolo *University College London, UK*

Exhibits
Rose Gamble *U. of Tulsa, USA*
Rick Kazman *SEI & U. of Hawaii, USA*

Doctoral Symposium
Gail Murphy *U. of British Columbia, Canada*
Kumiyo Nakakoji *U. of Tokyo, Japan*

Most Influential Paper
David Notkin *U. of Washington, USA*

New Faculty Symposium
Leon Osterweil *U. of Massachusetts, USA*

Midwest Consortium
Matthew Dwyer *Kansas State U., USA*

High Schools Liaison
Kenneth J. Goldman *Washington U., St. Louis, USA*
Christine Roman *St. Louis Science Center, USA*

Student Volunteers
Bruce McMillin *U. Of Missouri - Rolla, USA*

Sponsorship
Sol Shatz *U. of Illinois, Chicago, USA*

Treasurer
Judith Stafford *Tufts U., USA*

Publicity, Design & Advertising
Daniela Damian *U. of Victoria, Canada*
Will Tracz *Lockheed Martin, USA*

Proceedings
Frank Maurer *U. of Calgary, Canada*

Local Arrangements
Chris Gill *Washington U., St. Louis, USA*

Webmaster
Amy Murphy *U. of Rochester, USA*

Press Liaison
Alexander Wolf *U. of Colorado, USA*

ICSE is the premier forum for researchers, practitioners and educators to present and discuss the most recent ideas, innovations, trends, experiences, and concerns in the field of software engineering.

Software Everywhere is the theme of ICSE 2005. It acknowledges the increasingly important role software plays in the life of our society through the technology that sustains it. The theme also highlights the growing level of responsibility our profession and its members are expected to assume. As such, an important goal of this meeting will be to reach out to other disciplines that have an impact upon or benefit from software engineering know-how.

Lasting Impact on our profession and the society at large is the overarching goal that shaped the programmatic agenda for ICSE 2005. Format changes, novel initiatives, exceedingly high expectations, an exceptionally talented team, and an unprecedented level of support by the local corporate community are some of the ingredients bound to facilitate a fertile exchange of ideas and experiences likely to affect the professional life of each participant. The conference will offer an exciting program of events, including keynote talks by leaders in the field, invited talks along specialized themes, tutorials, workshops, and technical paper presentations on innovative research, the cutting edge of practice, and new developments in software engineering education.

High Quality Submissions are invited for papers describing original unpublished research results, meaningful experiences, and novel educational insights. Proposals for tutorials, workshops, research demonstrations, exhibits, and poster presentations are also welcome. Topics of interest include, but are not restricted to:

- Software requirements engineering
- Software architectures and design
- Software components and reuse
- Software testing and analysis
- Theory and formal methods
- Computer supported cooperative work
- Human-Computer Interaction
- Software processes and workflows
- Software security
- Software safety and reliability
- Reverse engineering and software maintenance
- Software economics
- Empirical software engineering and metrics
- Aspect-orientation and feature interaction
- Distribution and parallelism
- Software tools and development environments
- Software policy and ethics
- Programming languages
- Object-oriented techniques
- AI and Knowledge based software engineering
- Mobile and ubiquitous computing
- Embedded and real-time software
- Internet and information systems development

Opportunities for Professional Engagement are available at all levels. Workshops, tutorials, research demonstrations, exhibits, and paper presentations offer possibilities for training and technological assessment. The new faculty and doctoral symposia are designed to help young researchers gain a head start in their chosen profession. College students can serve as volunteers and enjoy an intellectually enriching experience. For the first time, high school students will be given the chance to exhibit work as part of the conference.

The heartland of America and St. Louis welcome the conference in the elegant setting of the Adams Mark Hotel on the Mississippi riverfront and in the shadow of a monumental feat of engineering, the St. Louis Arch. The starting point for the historical Lewis and Clark expedition and the cradle of jazz, the region offers visitors a wide range of tourist and entertainment opportunities for both individuals and families with children.

Platinum Level Support has been extended to ICSE 2005 by the following members of the St. Louis community: Boeing, Emerson, Edward Jones, MasterCard International, Monsanto, SBC, and Washington University. Conference organizers gratefully acknowledge their generosity.

Conference Information will be available on the conference web site, which has been designed to be a living document offering up-to-date news on all conference events, submission instructions and deadlines, hotel information, registration, tourist information, travel, etc. The continuously updated web pages will help you plan your involvement in the conference. We encourage you to leave your own mark on this important event.

Important Submission Dates

Research, experience, and education papers	1 September	2004
Tutorial and workshop proposals	4 October	2004
Doctoral symposium	6 December	2004
Research demonstrations and posters	7 February	2005

Seapine Software™
changing the world of software development

Developers love creating code... Managers crave process control... Bridge the gap with Seapine CM.

Software development is a team effort with developers, testers, and management all working toward one goal – delivering the highest quality product on time.

Built on award-winning TestTrack Pro and Surround SCM, Seapine CM brings structure to source control and issue management, improving communication while accelerating product development.

Seapine CM helps your team...

Define custom change request workflows, putting you in control of who makes changes and who authorizes the closure of issues.

Associate source code changes with defects or change requests.

Gain a thorough understanding of how much work remains before project completion.

View complete audit trails of what changed, why, and by whom.

Understand how close you are to release—how many issues are open, how quickly are you closing them, how many are re-opened?

Successful team-based development requires the proper process supported by the right development tools. Tools that are flexible, easy to use, secure, and scalable—like Seapine CM.

Features:

Complete source code control with private workspaces, automatic merging, role-based security and more.

Comprehensive defect management—track change requests, bug reports, feature requests and more.

Fast and secure remote access to your source files and defects—work from anywhere.

IDE integration with JBuilder, Visual Studio, Dreamweaver, and other leading development tools.

Advanced branching, triggers, email notifications, fully configurable workflows, and customizable fields put you in complete control of your process.

Open interfaces—triggers, email notifications, SOAP support, XML data exchange, and ODBC clients.

Scalable and reliable cross-platform, client/server solution supports Windows, Linux, Solaris, and Mac OS X.

Achieve major improvements in software development performance through better tool integration and process automation. Manage defects, development issues, and change requests with award-winning TestTrack Pro and gain complete control over your source code and change process with Surround SCM. Seapine's integrated change management tools are feature rich, highly scalable, Web enabled, and cross platform. Streamline your development process with Seapine CM and help your team deliver quality software products on time, every time.

Learn more about the Seapine CM suite at www.seapine.com or call 1-888-683-6456

Surround SCM Source Control

TestTrack Pro Defect Management

QA Wizard Automated Testing

software development 13th annual product excellence award

Innovative Technology for Computer Professionals

Computer

March 2004, Volume 37, Number 3

Cover design and artwork by Dirk Hagner

ABOUT THIS ISSUE

Some serious challenges are emerging in the computer design industry: How can we design hardware to run faster than worst-case design would normally allow? Can we run chips faster than normal, boosting the clock until the logic implementing the CPU starts to fail, and then backing off? Are there forms of circuit-level speculation that improve performance? The articles in this issue address these and other related questions, providing a comprehensive look at the converging factors that are bringing about fundamental changes in microarchitecture design.

Computer

IEEE Computer Society: http://computer.org
Computer: http://computer.org/computer
computer@computer.org
IEEE Computer Society Publications Office: +1 714 821 8380

OPINION

NEWS

MEMBERSHIP NEWS

COLUMNS

DEPARTMENTS

Membership Magazine of the IEEE Computer Society

IEEE

ARTICLE SUMMARIES

Billion-Transistor Architectures: There and Back Again

pp. 22-28

Doug Burger and James R. Goodman

In September 1997, *Computer* published a special issue on billion-transistor microprocessor architectures. Comparing that issue's predictions about the trends that would drive architectural development with the factors that subsequently emerged shows a greater-than-predicted emphasis on clock speed and an unforeseen importance of power constraints.

Of seven architectural visions proposed in 1997, none has yet emerged as dominant. However, as we approach a microarchitectural bound on clock speed, the primary source of improved performance must come from increased concurrency. Future billion-transistor architectures will be judged by how efficiently they support distributed hardware without placing intractable demands on programmers.

Metamorphic Programming: Unconventional High Performance

pp. 30-38

Peter M. Maurer

A programming methodology that violates most of the rules of good programming has shown spectacular reductions in simulation times on several benchmarks. Applying this technique in logic-level VLSI circuit simulation also improved simulation performance. For a new VLSI circuit, faster simulation translates into faster time to market, so even the most peculiar programming type is worth exploring if the carrot is increased performance.

Discovering efficient and effective metamorphic programming techniques across a range of problems outside simluation will require a concerted effort across the software community. The most important problem is the lack of metamorphic constructs in mainstream high-level languages.

Reliable and Efficient System-on-Chip Design

pp. 42-50

Naresh R. Shanbhag

To increase processor performance, the microprocessor industry is scaling feature sizes into the deep submicron and sub-100-nanometer regime. The recent emergence of noise and the dramatic increase in process variations have raised serious questions about using nanometer process technologies to design reliable, low-power, high-performance computing systems.

The design and electronic design automation communities must work closely with the process engineering community to address these problems. Specifically, researchers must explore the tradeoffs between reliability and energy efficiency at the device, circuit, architectural, algorithmic, and system levels.

Going Beyond Worst-Case Specs with TEAtime

pp. 51-56

Augustus K. Uht

Virtually all engineers use worst-case component specifications for new system designs, thereby ensuring that the resulting product will operate under worst-case conditions. However, given that most systems operate under *typical* operating conditions that rarely approach the demands of worst-case conditions, building such robust systems incurs a significant performance cost. Further, classic worst-case designs do not adapt to variations in either manufacturing or operating conditions.

A timing-error-avoidance prototype provides a circuit and system solution to these problems for synchronous digital systems. TEAtime has demonstrated much better performance than classically designed systems and also adapts well to varying temperature and supply-voltage conditions.

Making Typical Silicon Matter with Razor

pp. 57-65

Todd Austin, David Blaauw, Trevor Mudge, and Krisztián Flautner

Voltage scaling has emerged as a powerful technology for addressing the power challenges that current on-chip densities pose. Razor is a voltage-scaling technology based on dynamic, in-situ detection and correction of circuit-timing errors. Razor permits design optimizations that tune the energy in a microprocessor pipeline to typical circuit-operational levels. This eliminates the voltage margins that traditional worst-case design methodologies require and lets digital systems run correctly and robustly at the edge of minimum power consumption.

Occasional heavyweight computations may fail and require additional time and energy for recovery, but the optimized pipeline requires significantly less energy overall than traditional designs.

Speeding Up Processing with Approximation Circuits

pp. 67-73

Shih-Lien Lu

Current microprocessors employ a global timing reference to synchronize data transfer. A synchronous system must know the maximum time needed to compute a function, but a circuit usually finishes computation earlier than the worst-case delay. The system nevertheless waits for the maximum time bound to guarantee a correct result.

As a first step in achieving variable pipeline delays based on data values, approximation circuits can increase clock frequency by reducing the number of cycles a function requires. Instead of implementing the complete logic function, a simplified circuit mimics it using rough calculations to predict results. The results are correct most of the time, and simulations show improvements in overall performance in spite of the overhead needed to recover from mistakes.

THE END OF SCIENCE REVISITED

In "The End of Science Revisited," (Jan. 2004, pp. 37-43), John Horgan raises a number of interesting points that we must consider in our profession.

I suspect that a parallel article could be found that was written a century ago discounting the possibility of human flight and scoffing at ever reaching the moon. Which is not to say that John Horgan's assertions are wrong or will be proven wrong over time.

Where I most disagree with Horgan is in his conclusion: "Science is never more dangerous than when it seeks to tell us what we are, what we can be, and even what we should be." Science does speak to these first two issues in many ways. It is not the final voice on the subject, but it is not a voice to be rejected.

Where science is most dangerous, in my opinion, is where it assumes that its results will be benign. One example is useful: If we accept Horgan's "scientific theological atheism" and assume that machine intelligence will not emerge, we could pass the "event horizon" that some have anticipated and not be able to get the genie back in the bottle. We only need be agnostic on this point to realize that we might need to consider the implications of technology that could pass beyond our control before we reach the projected timeframes for that event.

I encourage *Computer* to facilitate this dialogue with views from various perspectives, perhaps tracking the related articles and making them accessible on a Web site. I've posted some related information at http://acadweb.snhu.edu/Isaak_James/future.

Jim Isaak
Manchester, N.H.
CS2004@JimIsaak.com

Regarding John Horgan's musings about whether science is at its end, my response is that it is difficult to say that we know how much we don't know. I could leave it at that, particularly because of the failings of past projections by eminent people.

The author shows a degree of speculation and "faith" when he suggests that we are scraping the bottom of the barrel. He further claims that, "Scientists need a certain degree of faith to bolster their confidence in the arduous quest for truth...."

First, "faith" is a much-abused word. It applies to concepts ranging from religious beliefs to trust in someone or something. Second, scientific investigation has nothing do with "faith." Scientists reflexively deny faith because failing to do so would violate the scientific spirit.

A scientist's speculations are not bounded by the rigor of the scientific procedure or evidence. Those off-the-cuff remarks are the ones on which journalists thrive.

Rephrasing the author's sentence, I would say that journalists are never more dangerous than when they seek to tell science what it is, what it can be, and even what it should be.

Prasad N. Golla
Plano, Texas
prasad.golla@alcatel.com

I did not read John Horgan's book, *The End of Science*, nor am I inclined to do so. However, I did read "The End of Science Revisited" and was appalled.

Physics and cosmology are still making major discoveries such as dark matter and energy in spite of major cutbacks in big physics budgets. Fusion research is still going on, and the promise of fusion reactors supplying energy is still very much alive despite large cuts in that budget as well.

The rain forest example of an intractable ecosystem to simulate is particularly bizarre. I would think that if someone knew the roles of the flora and fauna, it is reasonable to assume that it would be possible to make a simulation that would yield useful information and information granularity. I suspect that Horgan would then say, "The model's prediction of termite populations was off by 5 percent."

Can he find Nobel laureates to agree with him? Sure—and they can be way off the mark, especially when they step out of their respective fields. And why not "horganics" instead of "chaoplexity" to label so-called intractable systems? He really misses the boat in this area and falls prey to the Deepak Chopra syndrome. If a system is seemingly intractable, then the logic is that quantum phenomena must be a major component, and it will be forever beyond our keen and the province of religion.

Now let's talk about AI. Horgan's major point seems to be that we will never understand ourselves, let alone develop machines that think. This is very much a "the Earth is at the center of the universe" view.

There is pretty much universal agreement that intelligent machines are inevitable and will happen in this century. Will this solve the problems of humanity? Doubtful.

Most human problems require human solutions: people dealing with people on a one-on-one basis. If this is what Horgan is really trying to say, then his thesis is too simple to warrant an entire book and is hardly controversial.

Gary Feierbach
Belmont, Calif.
feierbach@comcast.net

I honestly enjoyed John Horgan's article, "The End of Science Revisited." However, at the risk of seeming mischievous, I do think that we live in times when new truths could well lie before our very noses, but we would not know them for what they are if our lives depended on it. Our society emphasizes the certainty of accepted answers, not the wisdom of acknowledging those questions to which answers are not

readily forthcoming—the ultimate seeds of science.

We are so blinded by the view that science is relentless progress without any painful revisions of viewpoint that we are as susceptible as ever to unique surprises. When humanity truly desires to entertain a new thought and revise the conventional wisdom, we will progress. Indeed, the new thoughts may already be among us, remaining unrecognized.

Until then, we could be socked repeatedly in the jaw with a cold dead salmon and be none the wiser for it.

Rationality has its limits, not least of all the blind spot of our founding premise. When science recovers the will to ask new questions that challenge the limits of popular understanding, we will move forward. Until then, we remain blinded by the conceit of living off the foresight and courage of thinkers who came before us. They are the best of times and the worst of times. Creativity waits in the wings, but the times are not yet receptive to that.

Truly, it is that simple.

Kingsley Jones
Sydney, Australia
krwjones@bigpond.net.au

IT EMPLOYMENT PROSPECTS

I was surprised by the inclusion of an article that uses the ITAA as an authoritative source in *Computer*'s January issue (Fred Niederman, "IT Employment Prospects in 2004: A Mixed Bag," pp. 69-77).

The ITAA is a trade organization with the single purpose of advancing the interests of its members, which include many companies that want special treatment in the areas of hiring and fast-tracked importation of cheap labor and are willing to buy legislation to get it. The ITAA was proclaiming a huge IT labor shortage while companies were busily getting rid of their IT employees. Then the ITAA claimed that a huge shortage of IT labor was imminent while companies continued to dump more workers and the recession deepened. Even the computer trade rags have been so embarrassed by repeating claims from the ITAA that they now qualify their citations.

Anyone interested in the veracity of the ITAA and its president, Harris Miller, need only do a Web search on "harris miller electronic voting." The ITAA will say anything as long as they are paid to say it—that is what they do for a living. That kind of source, unless used for documentation of its nonsense, does not belong in *Computer*.

Terrence Vaughn
Garretson, S.D.
t.vaughn@computer.org

CLOCK CYCLE ERROR

The Industry Trends column in *Computer*'s January issue (Steven J. Vaughan-Nichols, "Vendors Go to Extreme Lengths for New Chips," pp. 18-20) includes the following quote: "The best way to increase the number of executed instructions per clock cycle is by increasing a chip's frequency." Last I checked, a clock cycle was the inverse of frequency. So how is it that we're going to increase the number of executed instructions per clock cycle?

This leads to the following conclusions: the source is in error, the writer did not catch the error, and the editor did not catch the error.

I think we can do better to maintain *Computer*'s quality.

Richard L. Lozes
Pleasanton, Calif.
richard_lozes@amat.com

WSIS ESSAY

The topic of the December 2003 The Profession column is one that needs to be aired more often (Neville Holmes, "The Digital Divide, the UN, and the Computing Profession," pp. 144, 142-143). Personally, I don't hold out much hope that people, organizations, and governments that benefit from the current inequities in the use of digital technology can be counted on—or persuaded—to correct them. I think that the needed changes will have to come from the bottom up, with some assistance from people of conscience—folks like George Soros, for example—who also have power and resources to help.

It needn't take much to get a good start. Take a look at the Grameen Foundation (www.grameen-info.org), which has been making microloans to poor people for a few decades now. This organization has helped the communities it serves make considerable progress while creating an entire business community around the idea of microcredit. Its US "branch" is currently starting up a technology center (www.tech.gfusa.org) that will start by helping microcredit organizations get the basic automated infrastructure that commercial banks take for granted.

Although I don't have any good links to offer, a number of other organizations are working specifically to develop a basic communication and information infrastructure for poorer countries and communities. Maybe it would be worth identifying these organizations in another article and commending them to professionals wishing to help.

Don Dwiggins
Northridge, Calif.
d.l.dwiggins@computer.org

Neville Holmes responds:

I appreciate receiving this information. I have added the two URLs, plus another pointing to an *Economist* article on microcredit in India, to the links I provide with The Profession column, which include other examples of providing low-level technical help to poor people (www.comp.utas.edu.au/users/nholmes /prfsn/2003.htm#3Dc). And if any reader is willing to write a 2,000-word essay on "Microcredit and the Computing Profession," I would be delighted to consider it.

We welcome your letters. Send them to computer@computer.org. Letters are subject to editing for style, clarity, and length.

32 & 16 YEARS AGO

MARCH/APRIL 1972

MICROPROGRAMMING (p. 17). "We have passed through a period when for technological reasons there was real meaning to the concept of machine instructions interpreted by a microprogram in read-only storage. The period was the '60s and the reasons were that we could get read-only memory that was exceedingly faster than core."

Y2K (p. 42). "I believe that the average computer of the year 2000 will:

- be an interpretive engine capable of executing directly one or more higher level languages,
- have wider words than today's machines, possibly with as many as six addressing fields per instruction,
- be predominately a stand alone machine with provision for occasional remote accessing of data,
- have no central registers,
- probably be a decimal machine,
- have a small wired in (microprogrammed?) operating system,
- have word by word protection and data description,
- be a monoprocessor doing its own I/O,
- most probably be privately owned and monoprogrammed."

MANAGEMENT (p. 48). "Evidence was found that soundness of management policy was what makes the difference between a good and poor documentation picture. The problem was not so much the attitude of the programmer as it was the attempts made by those in middle management to deal with the problem of documentation in their own way."

FLEXIBLE DISC FILE (p. 59). "Memorex Corporation has announced their new 650 Flexible Disc File, a compact, direct access unit which enables OEM's to greatly simplify the storage and handling of digital information with much greater reliability and higher performance than possible with cassettes or any other comparably priced file on the market today.

"… The unit has a capacity of 1.5 megabits, track to track access time of less than 50 milliseconds and a data rate of 200 kilobits per second."

INSTANT INVOICES (p. 63). "Customers who buy lawn equipment from Marr Brothers, Inc., don't have to wait long for their invoices. Owner James Marr says the firm's IBM computer, located behind the sales counter, has eliminated the need for time-consuming, hand-written paperwork on each sale.

"When a customer first visits the store, his name, address and pricing information about his account are keyed onto a punched card. On that and subsequent visits, the customer's card, along with cards describing the parts he purchases, are entered into the IBM System/3 Model 10. The computer automatically prepares a complete invoice."

MARCH 1988

NEURAL NETWORKS (p. 9). "Both the literature and the number of professional society meetings focusing on artificial neural systems are growing at an amazing rate. …

"… Although the field of artificial neural systems has roots going back over 25 years, there currently is no consensus of what is important to study or how to go about studying it."

SPEECH RECOGNITION (p.13). "Because the brain has already implemented the speech recognition function (and many others), some researchers have reached the straightforward conclusion that artificial neural networks should be able to do the same, regarding these networks as a panacea for such 'natural' problems. . . . What these people fail to realize is that *we may not yet have discovered what biological neurons and neural systems are like*."

STARTING SALARIES (p. 125). "Starting salaries for data processors have reached an all-time high in 1988, … The average increase over 1987 is 4.2 percent.

"Some specific starting-salary ranges are $37,000–$44,000 for project managers at medium-size installations, … $61,000–$81,000 for management information systems directors at large installations, … $20,000–$25,000 for programmers at small installations, … and $36,000–$44,000 for systems analysts at large installations …"

SDI (p. 125). "The US Strategic Defense Initiative Organization has awarded the contract for the National Test Bed to Martin Marietta Corp. The $500-million contract is for five years to develop a national network of supercomputer and simulation facilities designed to evaluate the feasibility of the Strategic Defense Initiative, commonly called 'Star Wars.'

"… Members of Computer Professionals for Social Responsibility, a nonprofit public interest organization of people in the computing field, have studied the project's official request for proposals, as well as available news stories.

"… In summary, the organization, based in Palo Alto, Calif., feels that 'the National Test Bed is a waste of taxpayers' money.'"

MAINFRAMES (p. 136). "Unisys has added two new models to its 1100/90 family of mainframe computers."

"… The single-processor 1100/91 Model II SV uses 256K RAM chips and comes in 8M-byte or 16M-byte memory units. Including operator console and system control software, it costs $1,429,000.

"The dual-processor 1100/92 Model II SV costs $2,605,000. It features a maximum system memory of 32M bytes."

Editor: Neville Holmes; neville.holmes@utas.edu.au.

The Zen of Overclocking

Bob Colwell

Every once in a while, something comes along in the computer industry that really surprises me. The first time this happened was in the early 1980s, when the first personal computing stores showed up in shopping malls. The second time was when I took the first Internet browser out for a spin and found a new universe of Web sites. The third occasion was using Altavista. I'm still amazed by how well search engines work.

A new "surprise" has been gradually creeping up on me, and I don't know whether to admire it or heap scorn on it. So I'll do both, by turns. It's called *overclocking*—driving your computer faster than its specifications allow.

HOT RODS, MUSCLE CARS, INSANITY

In the 1950s and 1960s, one cool thing for high school or college students to do was to "soup up" their automobiles. While I wasn't personally afflicted with this particular malady, it was part of the general culture, at least in the US. The symptoms included an oversized, overpowered American car, an air scoop for the engine, a new hood with a hole for the air scoop, a wing-like deflector on the back (to keep the back end pushed against the ground while driving at supersonic speeds), replacement of the muffler with chrome pipes that resonated most alarmingly, a metallic green or blue paint job, reverb springs on the AM radio and 8-track tape player, and some kind of cheesy fire motif adorning the fenders.

Overclocking is a large, uncontrolled experiment in better-than-worst-case system operation.

In an era when pollution and fuel economy were blithely ignored, my crazy neighbor even put a nitro engine in his car so that he could leave even more rubber on the pavement than a normal overpowered V-8 allowed. If the crazy part isn't obvious yet, consider this: This beast burned fuel like a brush fire, fuel that cost three times as much as gasoline and was available from only one station in the entire city. To the extent that this assemblage was intended as a babe magnet, there is some question about the wisdom of limiting its range to a one-mile radius around a single filling station.

Why would people do this? Too much time and money on their hands is one plausible explanation, and it's at least a prerequisite for any other explanations. But there is also something to admire here—people who modify their vehicles this extensively are obviously not afraid of technology. They get their hands dirty. They want to know how things really work. They're not content to treat technology as a closed black box; in fact, they don't really trust technology that must be approached in that way. Except for their lack of parsimony, and their willingness to take naïve, unjustified chances with technology, they sound like engineers.

PC TWEAKING

Overclocking has been around since the early 1990s, but it seems to be gaining popularity. I recently saw a magazine called *PC Modder* in my local supermarket. (Yes, not a specialty bookstore, not the local PC repair shop—the *supermarket*.) It has step-by-step tutorials on everything from CPU, chipset, DRAM, and video card overclocking techniques to special cooling arrangements (yes, including compressor-based refrigerators) to painting orange and red flames on the sides of the computer skins that any 1950s hot-rodder would immediately recognize. One person even installed the dashboard of a car into his PC and wired the gauges to reflect machine temperature, CPU load, and memory activity. There is actually a company that will help you perform such a tachometer modification on your PC (www.xoxide.com).

Just as with hot rods, there is a community here. Hot-rodders liked the idea that they not only could rebel against what the big companies were foisting on them, they could improve on it. As they say in *PC Modder*, "The allure of modding is pretty elementary. If some power is good, then more power must be even better. And if you look good in the process, well, that doesn't hurt, either … [A] strong sense of community is one of the reasons modding is so much fun."

A current TV commercial starts with one guy showing off the engine of his car to his friends, who all crowd around to peer inside. Then an Internet

service provider van pulls up in the next driveway, and the camera zooms out to show that the friends have all left the car and are now crowding around the new, faster computer next door, while the car owner is looking around bewildered. Maybe he should have painted flames on his doors.

A new book on this topic, *The Book of Overclocking: Tweak Your PC to Unleash Its Power* (Scott Wainner and Robert Richmond, No Starch Press, 2003), even comes with "attitude," with chapters like "What the Computer Industry Does NOT Want You to Know" and "Overclocking Versus Industry Hype: Technical Background."

OVERCLOCKING AND BETTER-THAN-WORST-CASE OPERATION

This issue of *Computer* spotlights what I call "better-than-worst-case" design. With normal worst-case design, any computing system is a conglomeration of components, operating within frequencies, power supply voltages, and temperature ranges that were set to simultaneously accommodate worst-case values of every single component. (Modern CPUs don't really do it quite this way anymore, but they once did, and it's easiest to think of worst-case design this way.)

Think for a moment about what that kind of worst-case design means. There can be hundreds of chips in a computer; there are tens of millions of subcomponents in each integrated circuit. The chips themselves were designed for worst-case operation of their cells, and then a manufacturing margin was added. Those chips were then designed into a system, and the margin process repeated.

When all is said and done, it's no wonder that these stacked margins result in noticeable headroom under nominal conditions. What are the odds that the power supply is drooping to its minimum acceptable value, each and every cell in each and every chip is performing to the minimally acceptable end of its guaranteed range, temperature is as high as is allowable, and the clock has drifted to its maximum? It's this headroom that lets the overclockers ply their trade and provides ammunition to the conspiracy theory-minded among them.

In effect, overclocking is a large, uncontrolled experiment in better-than-worst-case system operation. I'm reluctant to draw any strong conclusions from such an experiment, but ignoring it doesn't seem right either.

There are reasons to reconsider whether historical worst-case design will suffice in the future.

Overclockers can demonstrate computers apparently running stably at substantially higher clock rates than those that the chip vendors guarantee. Do they have a point? Has the computer industry "margined" itself into overconservatism?

Well, no—and yes. No, because there are very sound engineering reasons for why the manufacturers set the clock rates the way they do. The industry also understands something that the overclocking community seems to easily gloss over—a stable system that runs acceptably fast is a more useful and valuable goal than a system that runs 10 or 20 percent faster at the expense of random crashes and unpredictable data losses.

If the user only runs games, the threat of crashing may be small and the loss per crash acceptable. But that is not what the computing industry is all about. Computing has become a key part of the worldwide economic infrastructure. As Clayton M. Christensen pointed out in *The Innovator's Dilemma*: *When New Technologies Cause Great Firms to Fail* (Harvard Business School Press, 1997), once overall product development has matured to acceptable performance levels, reliability becomes the next requirement. After reliability is also achieved, the competition becomes purely economic.

LOOKING TO THE FUTURE

But there are reasons to reconsider whether historical worst-case design will suffice in the future, or even the modern statistical guardbanding equivalent. For instance, in the presence of random noise and a natural noise margin insufficient to reject it, even worst-case designs will still randomly fail. If that noise occurs too often, the system won't be acceptable to its potential buyers.

Articles in this issue of *Computer*, for example, point to the possible future of the industry, where increasing noise, decreasing noise margin, higher thermal power and higher leakage, lower signal strengths, and lower operating voltages will all converge to force a fundamental change in microarchitectures.

Today, these microarchitectures are designed for speed; correctness is achieved by default because the underlying circuits are expected to work correctly every time. In the future, that expectation may not be realistic. We may have to begin designing machines that check themselves as they go, automatically backing up and retrying operations that fail.

DEBUNKING OVERCLOCKING

I admire the "don't be afraid of technology" aspects of the overclocking culture. We *should* hold all technology purveyors accountable for their products and not meekly accept what is offered simply because those products embody complexity beyond our particular backgrounds. But not all aspects of the overclocking culture seem admirable to me.

For instance, there seems to be a dose of irrationality in some of the overclockers' thinking. The overclocking book says that the microprocessor industry's continuous performance spiral is really some kind of conspiracy: "… the computer industry prey[s] upon the fact that the vast majority of end users have limited hardware knowledge."

Overclockers say, "Instead of buying a new PC, just overclock the old one." But this is inconsistent—if the additional performance is good, then these critics should applaud the computer industry's record instead of labeling it as predatory. And if additional performance isn't intrinsically a good thing but is merely a side effect of industry hype, why go through all the headaches and risks of overclocking? Besides, as the overclocking book, the magazine articles, and some online forums that I have visited repeatedly point out (to their credit), "overclocking is not for everyone." In fact, it's not for the vast majority referred to above, for reasons overclockers themselves admit.

The overclocking community also seems peculiarly susceptible to "CPU relativism." This is a mental disorder in which all sense of objective reality is lost, and the only thing that matters is whether Intel's or AMD's chips are faster, or who announced the latest N-GHz breakthrough, and how that surely signals the coming of the Apocalypse. Exactly what it means to be faster is itself deemed worthy of limitless debate.

It's really weird. Consider: when you're sitting at home in front of your computer, in what possible sense does it affect your life if the person next door happens to have a computer based on the other company's CPU, which happens to run Irrelevant Benchmark Number 6 Epsilon faster than yours? I would have thought that "keeping up with the Joneses" only works if the Joneses notice.

The only premise mentioned in the overclocking book that makes any sense to me is the idea of staying one or two speed bins behind the leading microprocessor of the day and then overclocking to make up the difference. That is not to say I agree with this strategy; I'm just saying that nothing else makes any sense at all. Trying this two-bins-back plan might be tempting because the highest performing processors tend to command the highest average selling prices, not necessarily in linear proportion to the clock or performance differences. The two-bins strategy lets you pay less for your CPU, and you get equivalent system performance if you can get the system stable (at least to your own satisfaction) at the overclocked rates.

Most of the world's computing is done by people who are trying to accomplish something other than getting the high score on a game.

Since 1991, microprocessors have run with their internal clock rates set to some multiple of the bus clock rate. Intel's 66-MHz 486 chip had a 2:1 multiple. Then came 3:1, 5:2, and many others. Today's CPUs have dozens of allowable ratios to accommodate the increasing disparity between skyrocketing CPU clock speeds and non-skyrocketing memory and I/O subsystem speeds.

In the early days, these ratios were set by various means accessible to BIOS or jumpers. But today, both Intel and AMD lock the ratios at manufacturing time. Overclockers believe this is part of some overall conspiracy against them, but the truth is, they don't constitute a large enough block of buyers to really influence business plans that way. Bus locking is really aimed at preventing "gray market" scams, in which the scammers buy up large lots of CPUs, remark them as faster parts, and then resell them at the higher prices that those faster speeds command.

Some fraction of those re-marked parts won't function well at the higher frequency, generating service calls to the original manufacturer, who then must break the bad news to the gray market buyer that he has no service warranty because the product was altered. It's bad for business: Both the buyer and the vendor are unhappy.

SCIENCE AND SAFETY MARGINS

Compare the seat-of-the-pants, maybe-it-will-work approach of the overclockers to the engineering challenge confronting Intel and AMD. First, note that this challenge isn't just the flip side of the overclocker's coin. Chip manufacturers must design and produce tens or hundreds of millions of chips; overclockers only worry about one. Manufacturers must set a quantifiable reliability goal, and no, it's not "zero failures, ever." That would be an unreachable—and not very productive—target because hitting it would require avoiding cosmic rays. Even at sea level, that would require more meters of concrete than any laptop buyer is going to find attractive. And even then, the concrete would only improve the odds. It would remain a statistical game.

The statistical target chosen for overall machine reliability in the face of cosmic rays is also used as a rough calibration for the required reliability of the silicon in general. To obtain a chipwide reliability model, production and design engineers must apply probability distribution functions to circuit speed paths, in combination with all other factors that could affect the speed. They then combine this model with other models for expected operating conditions, types of code being run, and the chances that any errors generated will be reflected in any program output (a tree that falls in a silicon forest doesn't make a sound if no one hears it.)

These models have become quite sophisticated, and they are undoubtedly considered closely guarded secrets within each company. These statistical approaches work for these companies because they have shipped a large volume of parts over a long period of time on silicon processes they developed themselves; therefore, they have the necessary experience base to inform the models.

Overclockers have none of those things—not the science, not the math models, and not the data required.

They don't even have access to the models the vendors use, so they can't know what guardbands have been built in. All overclockers can do is try it and see if it seems to work. For running games, this method may work, and it should be fun and educational. But they shouldn't talk about industry conspiracies.

Most of the world's drivers would not benefit from having nitro-breathing, flame-sporting vehicles; they just want to get from here to there and back again, reliably and economically. And most of the world's computing is done by people who are trying to accomplish something other than getting the high score on a game. For them, fast-yet-reliable is the right target for computer designers and chipmakers, and overclocking is anathema.

You've probably seen those fighting robots on TV. Engineering something that will either destroy or be destroyed may seem familiar to our colleagues in the military, but it does call for a different approach than the long-term view needed in computing.

Here's a glimpse of this different mindset: "The main weapon drive motors are 90 VDC units from a surplus buttocks exerciser. But operating components at their rated parameters is a glaring sign of inexperience in this sport, so we plan on laying at least 120 V into these babies. And they're going to like it!" (www.teamdelta.com/hazard/hazjan.htm). I don't even want to think about what this exerciser was originally intended for, or why the implied exercise is ever warranted, but it doesn't surprise me that the equipment ended up as surplus.

If you don't floss your teeth, they won't necessarily rot away. The vast majority of car trips do not include any metal bending, so why wear seat belts? And why not smoke? Not all smokers get cancer. Or you could adopt Oscar London's compromise, "If you smoke, why bother wearing a seat belt?" And some rock musicians from the 1960s are still alive, so maybe all those drugs are really beneficial, acting as some kind of preservative.

As for me, well, I'm an engineer, and I live in a statistical world. I'm going with the odds. ■

Bob Colwell *was Intel's chief IA32 architect through the Pentium II, III, and 4 microprocessors. He is now an independent consultant. Contact him at bob.colwell@comcast.net.*

Voice Authentication Speaks to the Marketplace

Steven J. Vaughan-Nichols

Although the hype over biometric security has cooled down as users have gained exposure to the technology, one particular approach has started to become increasingly popular: voice authentication.

Individuals trying to make a purchase with a credit card, access a protected computer system, or retrieve account information from their bank can speak into a telephone and have their voice identified by a voice-authentication system to verify they are who they claim to be.

One US railroad uses ScanSoft's SpeechSecure voice-authentication software to ensure that the customer releasing a rail car after it's been unloaded is authorized to do so, noted ScanSoft spokesperson Marie Ruzzo.

Judith Markowitz, president and founder of J. Markowitz, Consultants, a voice-biometrics consultancy, said, "2004 has been an interesting and active year for security as a whole, including voice authentication, because the US government is distributing considerable homeland-security funding. And so, I'm seeing a lot of action not only for voice but for other advanced biometric security uses."

Jackie Fenn, vice president at Gartner Inc., a market research firm, said government will be the primary voice-authentication adopter to address its many security concerns. She said corporate adoption will continue to grow

slowly until biometric readers are routinely embedded in [private-branch-exchange phone systems].

A growing number of voice-authorization products are appearing, such as Courion's PasswordCourier, Nuance Communications' Nuance, Vocent Solutions' Confirmed Caller, Voice.Trust's Voice.Trust server, and Voicevault's Voicevault.

However, concerns in such areas as security and accuracy may be significant hurdles to the technology's widespread adoption.

THE TECHNOLOGY

Essentially, voice-authentication systems capture and digitize speakers' voices. The basic equipment is a microphone or telephone to input speech, an analog-to-digital converter to digitize the spoken words, a high-powered computer, and a database to store voice characteristics.

Typically, these systems match a voice's harmonic and resonant frequencies, as well as the way the speaker pronounces phonemes—a language's smallest distinctive sounds—against an authorized user's digital voiceprint. The voiceprint is created when the authorized user enrolls in the authentication system, and it is subsequently stored as a digital file in a database.

The system calculates a score that indicates how closely the spoken voice matches the stored voiceprint for the person the speaker claims to be.

Alvin F. Martin, a mathematician at the US National Institute of Standards and Technology and an organizer of NIST's annual evaluation of speech-recognition programs, noted, "The increase in processing power in modern computers has helped make voice authorization more effective."

In addition to chips that can quickly process the large amounts of information involved in voice authentication, the systems need huge memories to store the data and pattern-matching technologies to compare live speech with stored voiceprints.

SPEAKING OF ADVANTAGES

As a biometric identifier, voice authentication has much to offer, said Steve Bittner, vice president of development for Convergys, a business and call-center services company.

For example, the technology permits re mote authentication, unlike other biometric approaches such as fingerprint or iris scans. A user can enroll in and work with a voice-authentication system from a remote location via a telephone.

Also, many users are more comfortable identifying themselves by speaking than by submitting to fingerprint or iris scans, which are frequently seen as invasive.

Voice authentication can also reduce the cost of handling customer-service calls, according to Bittner. Voice-authentication systems with speech recognition could verify a speaker's identity, determine the spoken reason for the call, and forward the call to the appropriate service. This would save money by reducing the number of call-center employees.

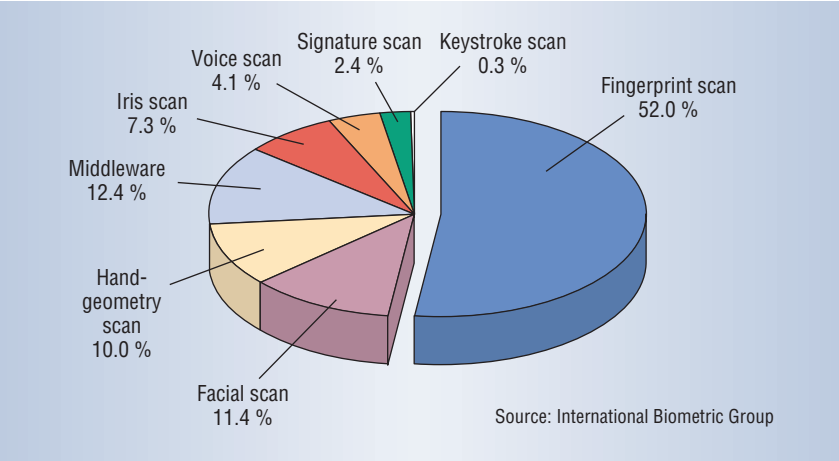


Figure 1. Voice authentication represents only a small part of the overall biometrics market.

Bank of America senior vice president Tim Wishon noted that his company used this approach to combine about 4,000 toll-free numbers into a much less expensive system using just one toll-free number.

USING VOICE AUTHENTICATION

Various government agencies use voice authentication for security purposes, such as ensuring that only authorized users have access to computer files or buildings.

There is also some commercial adoption of voice authentication, primarily by financial services companies. Markowitz said the technology is a good fit for the industry because of its security requirements and because customers like to perform many transactions and activities, such as verifying account balances and resetting passwords, via telephone.

In addition, merchants are using voice authentication for telephone-based, credit-card transactions, to reduce the risk of fraud by verifying that the voice on the line is that of the card's owner.

Voice authentication is particularly well suited for remote network and system access, employee timesheet record keeping, and other applications that require callers to use the same basic identification process, such as providing their mothers' maiden name, during authentication.

Voice authentication is also good for security processes that must identify many individuals because remote users need only a telephone and companies thus don't have to buy expensive equipment for them.

HURDLES

To inspire confidence and encourage more widespread adoption, voice authentication must overcome several obstacles.

For example, said Michelle M. Shen, consulting manager of ePolymath Consulting, a biometric consultancy, "The time [required] to verify a customer can be very long. Voice templates are so much larger than other kinds of biometric information. For example, data associated with a fingerprint may take up only 10 Kbytes, while a voiceprint typically takes up from 500 Kbytes to 1 Mbyte. This makes fast database servers and quick-filtering software a must."

Also, organizations may not feel comfortable adopting such a new technology yet. In addition, biometrics, after receiving so much hype a few years ago, has been shown to be less than completely reliable at times.

Security

As with any technology that allows access to sensitive systems, there are concerns about whether hackers could compromise voice-authentication systems.

According to Markowitz, someone could play a recording of someone's voice to fool a low-end voice-recognition system. However, today's more sophisticated systems create detailed voiceprints that wouldn't match readily with a recorded voice.

Skilled human imitators, though, could still fool a pure voice-authentication system in many cases.

Consistent accuracy

Voice authentication is the least accurate biometric-security system, according to Gartner's Fenn.

In accuracy tests in lab settings, where environmental variables are controlled, voice-authentication systems compare favorably with other biometric approaches. In real-world use, though, behavioral and environmental factors such as background noise or changes in users' voices due to health, fatigue, or other causes reduce voice-authorization systems' accuracy.

"Voice characteristics vary with your age, your metabolic state, your emotional state, and all the ways you can say [various words]," said George Doddington, a speech-recognition expert and consultant to the US government. This makes relying on voice authentication alone as a security measure problematic.

"There are many breakdown points in voice authentication," explained ePolymath's Shen. A typical example occurs when people use a different type of phone for authentication than they did for enrollment. For example, a cellular phone used in traffic may produce somewhat different voiceprints than a high-quality wireline phone.

CLEARING THE HURDLES

Researchers are looking for ways to overcome voice authentication's obstacles.

Security

Some voice-authentication applications offer improved security via a two-

factor process, in which a user provides a voice sample along with another authenticating detail—such as a password or account number—in response to a question from the system. Voice authentication does the initial speaker identification and then speech recognition recognizes the user's answer to the contextual question.

The system rejects users who can't answer the questions correctly or refers them to a live agent.

Accuracy

Researchers are taking several approaches to improve voice authentication's accuracy.

For example, David Frogel, Courion's director of business development, said his company authenticates users more accurately based on a scoring system that compensates for a number of external factors that could change a speaker's voiceprint. These factors include whether a call is coming from an internal or external source or is affected by environmental variables such as background noise.

Although the voiceprint may be a bit different, the speaker's voice still must substantially match it for authentication to occur.

Speaker model synthesis. Nuance uses a speaker-model-synthesis approach to develop a machine-learning algorithm that identifies changes in a voice template—the stored master record of a voice—based on different equipment used. The system can recognize equipment by its transmission characteristics, such as cellular phones' narrower high and low voice-frequency ranges.

Over time, for all speakers, the system creates a transform voice template for each type of equipment used.

Model adaptation. Model adaptation is also a key to improving voice-authentication accuracy, said Kevin Farrell, ScanSoft's director of speaker verification. This approach creates a more accurate voice template by adjusting an individual's voiceprint parameters—such as harmonic and resonant frequencies—over time, based on additional voice data received from encounters with the speaker after enrollment.

Analyzing new factors. NIST's Martin noted that modern voice-authentication systems are analyzing and classifying speech factors other than harmonic and resonant characteristics, such as word combinations, accents, and additional linguistic and idiomatic features.

At the Massachusetts Institute of Technology's Lincoln Labs, senior staff member Douglas Reynolds is working on ways to analyze and classify previously unexamined acoustic information such as voice pitch, pauses, and pronunciation style.

Meanwhile, at IBM's Thomas J. Watson Research Center, Ganesh Ramaswamy and other researchers are developing their *conversational biometrics* technique by analyzing and classifying multiple types of speech-related information, including pronunciation and how speakers use sounds like "uh" when thinking of what to say.

According to Courion's Frogel, "Voice-based authentication is already gaining traction in the marketplace." And recent enhancements, such as user-friendly interfaces, are increasing the technology's popularity. However, products still need additional improvements, such as faster voice recognition and elimination of the need for users to repeat phrases many times.

Markowitz said that more effectively combining voice authentication and speech recognition might help address these problems.

Another challenge is smoothly integrating voice authentication with other systems. Currently, organizations must use proprietary middleware or custom integration. Vendors are exploring ways to solve this problem via standards such as the Common Biometric Exchange File Format, the Voice Extensible Markup Language, and programming interfaces such as BioAPI.

In the marketplace, said ePolymath's Shen, "Overall growth has been mediocre. If voice authentication is to grow, it's real potential will be in financial services."

As Figure 1 shows, the International Biometric Group consultancy found that in 2003, voice authentication accounted for only 4.1 percent of the $928 million biometrics market.

The IBG says voice authentication is just not accurate enough yet to increase its market share substantially in the near future. ■

Steven J. Vaughan-Nichols is a freelance technology writer based in Arden, North Carolina. Contact him at sjvn@vna1.com.

Editor: Lee Garber, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; l.garber@computer.org

Will Interoperability Problems Give IP Telephony a Busy Signal?

Neal Leavitt

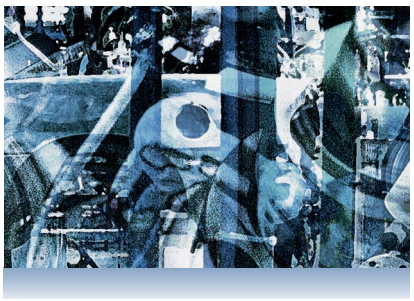
Internet telephony, also known as voice over IP, may finally be ready for prime time in 2004. An increasing number of IP telephony providers, traditional phone carriers, and now cable companies—including AT&T, Comcast, Cox Communications, Level 3 Communications, Net2Phone, Verizon, VoicePulse, and Vonage—are offering a growing range of corporate and residential services nationwide.

Internet telephony is attractive to customers because it uses lower-cost equipment and is less expensive to operate than traditional, circuit-switched telephony, thereby enabling carriers to charge lower rates.

In addition, the technology conveniently integrates data and voice services on the same carrier's IP network. Bundled voice and data are also less expensive than each service purchased separately.

And by combining the Internet protocol and telephony, voice over IP permits a broader portfolio of converged Web and voice services and applications, added Dan Dearing, vice president of marketing for NexTone Communications, an Internet telephony infrastructure vendor.

Most Internet-telephony traffic consists of international long-distance calls.



In 2002, Internet telephony accounted for 11 percent of all international long-distance traffic, according to a report by TeleGeography, a telecommunications-market research firm.

Industry research firms forecast steady growth in the IP-telephony market at least through 2007. For example, ABI Research estimates the market will increase from about $750 million this year to more than $8 billion by 2007. IDC predicts that corporate spending for IP-telephony equipment will grow 44 percent to $3.2 billion in 2004.

However, there are technical obstacles that could significantly impede the technology's marketplace growth. Providers' incompatible IP-telephony implementations make it difficult and expensive to move Internet calls between different carriers' networks. And as more calls shift to the Internet, these handoffs will grow in volume and complexity, further complicating the problem.

Researchers are thus working on ways to enhance interoperability.

BACK-END PROBLEMS

Many Internet-telephony providers rely on the traditional telephone network to carry their traffic. IP-telephony systems use soft switches, which are open application-program interface software that links traditional and Internet-telephony networks. They also typically employ media gateways, which use expensive digital signal processors (DSPs) to convert circuit-switched-based traffic originating and terminating on the traditional phone network to and from IP traffic.

Problems with standards

A key Internet-telephony interoperability problem involves the different ways that vendors implement two key standards.

The standards. Two of the most important IP-telephony standards—the session initiation protocol and H.323—are at the heart of the interoperability problem.

SIP is an Internet Engineering Task Force signaling protocol for initiating, modifying, or terminating an interactive user session that includes multimedia elements such as voice, video, or gaming.

SIP handles communications requests from clients, which can be sent via various transport protocols, and responses from servers. After identifying the participating systems, SIP determines a session's communication media and media parameters, as well as the called party's interest in participating. SIP also enables sessions involving services such as call forwarding, caller authentication, Internet conferencing, and instant messaging.

H.323 is an International Telecommunication Union standard originally designed to promote compatibility in videoconferencing across disparate IP networks. Service providers have also used H.323 for Internet telephony

because it addresses call control and management, as well as gateway administration of media traffic, bandwidth, and user participation. The standard represents a very large protocol suite and thus requires extensive memory.

SIP was designed to be relatively simple and flexible and to enable application programmability and easy feature-set extension. H.323 is more rigid in its implementation but provides better session control and management.

H.323 and SIP systems aren't directly compatible.

Different implementations. Vendors have implemented SIP and H.323 in various ways for different reasons, such as to gain operational efficiencies or competitive advantages. In addition, vendors frequently interpret protocols differently or implement a standard's features before they've been approved. Standards also change over time to address market needs, leaving some users with older and different implementations.

Thus, not all H.323 systems can work together, and not all SIP systems can work together. Internet-telephony systems may reject calls because the sending device uses a different interpretation of a standard and can behave unexpectedly, such as by omitting mandatory fields from a protocol message or sending information via the wrong IP port.

These factors have hindered the deployment of next-generation network technologies that require interoperability both to work properly and to create a large enough market to generate desirable revenue streams.

Meanwhile, vendors are often slow to spend the time and money necessary to make their systems more easily interoperable, particularly if it entails giving up a possible competitive advantage.

TDM peering

Traditional phone systems carry multiple data streams by using time-division multiplexing. TDM puts multiple data streams on a single signal by separating the signal into many short segments. Each data stream is reassembled properly at the receiving end.

Some IP-telephony carriers send traffic outside their networks via TDM peering, which uses back-to-back media gateways. With this approach, carriers transform packet-based voice traffic to a circuit-switched format and vice versa at the gateways.

Converting Internet-telephony traffic to TDM and then back when it reaches the recipient eliminates the variables that make it difficult for the IP traffic from different systems to work together directly. However, TDM peering is cumbersome to engineer and the media gateways use DSPs, which are expensive.

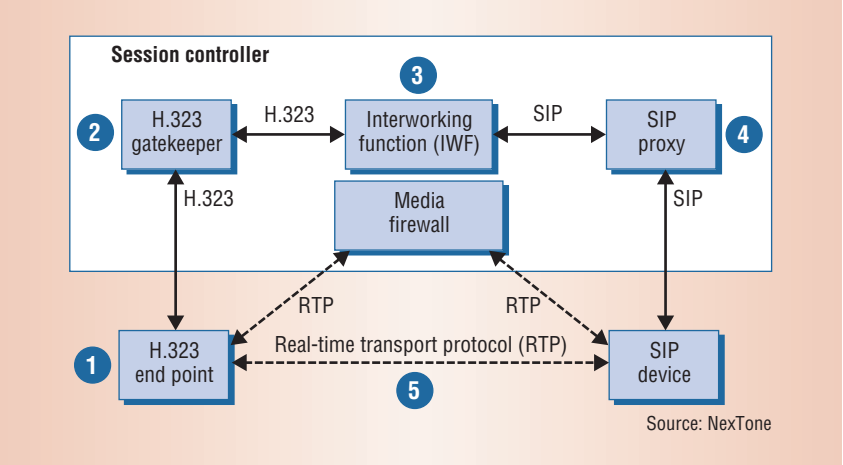


Figure 1. When an Internet-telephony call is made between an H.323 end-point device such as an IP PBX and a session initiation protocol device such as an IP phone, the H.323 gatekeeper lets the session controller communicate with H.323 end points. The SIP proxy lets the controller communicate with SIP devices. The IWF translates between H.323 and SIP and thus lets the different types of devices communicate. The media firewall provides security and controls access to a provider's network. The system uses RTP to carry the voice media.

PROVIDING INTEROPERABILITY

Researchers are working with session border controllers (SBCs) to overcome Internet telephony's interoperability problems.

"Session controllers have helped resolve [these] problems and enabled Internet-telephony carriers to peer [more easily and inexpensively]," said NexTone's Dearing.

How SBCs work

An SBC is a new breed of networking technology that provides routing, control, and security functions, as well as signaling interoperability and service quality, to manage real-time traffic between IP networks.

As Figure 1 shows, SBCs offer an H.323 and SIP interworking function (IWF) that uses software to translate between the protocols and thus provide routing services between devices.

When calls are placed between an H.323 and an SIP device, the SBC views each call as two legs: an ingress leg terminating on the IWF and an egress leg the IWF generates based on the protocol used by the remote destination.

SBCs thus eliminate the need to use TDM as a peering technology between IP networks. With session controllers, carriers have to convert packet-based traffic back to TDM only when sending or receiving calls from a traditional telephone network.

Companies such as Acme Packet, Jasomi Networks, Kagoor Networks, Netrake, and NexTone Communications produce SBCs.

Lower costs

Industry observers estimate that using SBCs can be up to 80 percent less expensive than TDM peering.

SBCs don't use DSPs, which makes the equipment less costly, and the controllers are much easier on network management than media gateways, explained Micaela Giuhat, Netrake's vice president of product management. Also, she said, SBCs can be provisioned in a couple of hours, while media gateways can require weeks.

Using SBCs

Even with SBCs, carriers must still test and tweak the points where their networks interconnect to maximize how efficiently they interoperate, particularly as new H.323 and SIP versions are released.

Carriers also must provide end-point information—such as vendor type, signaling protocol, and codec—so that the SBCs can determine which signaling changes are necessary to dynamically mediate between end points.

As carriers seek new sources of revenue, they are compelled to adopt more flexible business models that include peering with other providers.

"While the use of [IP-telephony] technologies reduces operating costs and provides new revenue-generating applications, it also creates new issues of … multivendor interoperability," Dearing noted.

If some carriers decide not to enable interoperability with other systems, "they will be out of business in five years," said Jeremy Duke, CEO of the Synergy Research Group, a market-research firm.

These factors will encourage carriers to begin standardizing the technology they use so that their systems work basically the same way and will interoperate without SBCs or other intermediary approaches, said Jerry Ezrol, technology leader for AT&T Labs' Voice over IP Development Group. Thus, he contended, the evolution to Internet-telephony interoperability is inevitable.

However, noted Eric Paulak, research vice president of the Network Services Group at Gartner Inc., a market research firm, "It's a migration and evolution in the marketplace. It won't happen overnight." ■

Neal Leavitt is president of Leavitt Communications, an international marketing communications company with affiliates in Paris; Hamburg, Germany; Bangalore, India; Hong Kong; and Sao Paulo, Brazil. He writes frequently on Internet and other technology-related issues. Contact him at neal@leavcom.com.

Editor: Lee Garber, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; l.garber@computer.org

Displaying Data in Thin Air

Two new techniques represent the latest approach to display technology: doing away with the screen. While unlikely to replace the desktop computer monitor, these thin-air displays could eventually be put to use in product showrooms, museums, military training facilities, corporate conference rooms, trade fairs, theme parks, and advertisements.

Chad Dyner, a graduate student at the Massachusetts Institute of Technology and chief executive officer for IO2 Technologies, has invented the Heliodisplay, which condenses the air above a video projector. The device then projects an image onto the condensed air, Dyner said.

IO2 has constructed proof-of-concept devices with 5-, 15-, 27-, and 42-inch screens. These can display two-dimensional images that hover above the projector. Because they are displayed on a surface that is not flat, the images appear 3D from a few feet away and can be seen from any surrounding position.

Dyner said he created IO2 Technology to license the technology to "one or more key players in the display market or companies that have the manufacturing capability to produce and distribute Heliodisplay."

Senior researcher Ismo Rakkolainen and Professor Karri Palovuori of Finland's Tampere University of Technology have developed the FogScreen, a display surface made out of a cloud of water vapor diffused into the air as a very dry fog. A projector can display images on the FogScreen.

"It appears dry to the touch, so it feels just like air," Rakkolainen explained. Viewers will thus see images they can walk through without getting wet. The water vapor is diffused between thick layers of emitted stable air, which keeps the fog thin and flat, enabling high-quality images, he added.

FogScreen technology, invented by two Finnish researchers, projects an image onto a display surface of water vapor diffused into the air as a very dry fog. Viewers can even put their hands through the images.

The first permanent prototype was installed in the Vapriikki museum in Tampere and has since been loaned for use in France. FogScreen is renting some devices and expects to begin sales soon.

Rakkolainen said, "This project started as a wild idea." The researchers formed FogScreen Inc. last year and are currently marketing the device.

Although Heliodisplay and FogScreen are interesting, it remains to be seen if they will be cost effective and will find an important use, said analyst Chris Chinnock of Insight Media, a display-industry research firm.

"Both are more likely to work as advertising or information screens in museums, public displays, entertainment centers, etc.," he explained. "These are specialty displays that need a novel experience and they need to be marketed and positioned as such." ■

—Linda Dailey Paulson

New Wireless Technology Provides Quality of Service

A new technology promises to guarantee service levels to wirelessly streamed digital content and thus may provide the basis for new digital entertainment products such as camcorders and other video and audio players.

The IEEE is working on standard 802.15.3a, which amends the organization's 802.15.3 ultrawideband (UWB) standard for high-rate, short-range wireless networks.

The amendment prevents the interruption of a data stream after a client device and the wireless network establish a link, explained Robert Heile, chairman of the IEEE's 802.15 Working Group and chief technology

officer of Appairent Technologies, a wireless-multimedia-product vendor.

IEEE 802.15.3a prevents interruptions by assigning a stream to a specific time slot that cannot be used by another stream until the first data set is fully transmitted. Networks using IEEE 802.15.3a also ensure continuous streaming by detecting noise from a source, such as a cordless telephone, affecting one transmission channel and assigning multimedia traffic to another channel not experiencing interference, thereby avoiding interruptions.

IEEE 802.15.3a also helps quickly detect lost packets via cyclic-redundancy-check techniques. "It is designed to get packets retransmitted fast enough so that you don't know you lost them," Heile explained.

The technology transmits data for up to 100 yards at 55 Mbits per second, quickly enough to efficiently handle multimedia traffic, he added.

Consumer-electronics and PC companies have been developing products using Wi-Fi wireless-LAN technology (IEEE 802.11) to handle multimedia streaming. However, Wi-Fi was designed to deal with bursty, rather than streaming, traffic. The technology also doesn't assign data to exclusive time slots.

In addition, Heile said, IEEE 802.11 uses only 30 to 40 percent of its available bandwidth, while IEEE 802.15.3a uses 80 percent.

Ian McPherson, an analyst with the Wireless Data Research Group, a market-analysis firm, said IEEE 802.15.3a will be important because "there will be a need for quality of service as digital content matures."

A catch with IEEE 802.15.3a is that it is a form of UWB. UWB sends the various pulses of a single transmission over a relatively large part of the radio spectrum, not just at a specific frequency or narrow frequency range as is the case with cellular-phone and other radio-based technologies.

Unlike the US, many countries don't permit commercial UWB transmissions because their regulations address only technologies that operate at a fixed frequency or a narrow frequency range.

Heile said IEEE 802.15.3a's success depends largely on whether many countries permit UWB in the future. If so, IEEE 802.15.3's only possible competition could come from the European Telecommunications Standards Institute's HiperLAN 2, a wireless standard with quality-of-service modes, McPherson noted. However, he added, "It's complex and difficult."

Heile said the first IEEE 802.15.3a products available will probably be dongles that establish connections between consumer-electronics devices and PCs. ■

—Linda Dailey Paulson

Computers That Don't Look Like Computers

They look like breadboxes, toy log cabins, old radios, gasoline cans, even dolls. But tucked inside the interiors of these items are the guts of a PC. It's part of a growing trend by enthusiasts of constructing computers in unlikely host objects.

For example, Henry Minsky, a software architect with Internet tool company Laszlo Systems, converted an old Teletype machine and an old wooden-case radio into PCs. Minsky said he wanted his computers to have an old-fashioned, retro look.

This PC, built within the case of an old radio, is an example of a new trend in which enthusiasts build computers in unlikely host objects, including breadboxes, gasoline cans, and even dolls.

Enthusiasts face numerous challenges in building these hidden computers. Frequently, space is limited within the host object, and the interiors were not designed to accommodate wiring, provide ventilation, and otherwise house PC components.

Minsky said he installed a large heat sink and a slow, quiet fan, as well as a DVD/CD drive in both of his PCs, which run Linux. He eliminated the need for external cables by using wireless Ethernet.

A key to the smaller versions of hidden computers are smaller motherboards, such as the mini-ITX, developed by Via Technologies and used in commercial machines by vendors such as Tranquil PC and Hush Technologies.

The 17-centimeter × 17-centimeter mini-ITX (about half the size of a typical motherboard) is available with 600 Mbytes to 1 Gbyte of memory and includes a CPU comparable to an Intel Pentium 3, which uses less power and produces less heat than today's powerful microprocessors, said Rick Clayton, Via's sales program manager.

Because they use slower CPUs, the hidden PCs generally aren't used for advanced computing. In addition, mini-ITX boards sacrifice flexibility. For example, users can't add more than one video or sound card, and the processor is soldered to the board. ■

—Linda Dailey Paulson

Reversible Computing May Improve Mobile Performance

Researchers are working toward developing one of the first reversible computers, a machine that promises to reduce energy consumption and thereby enable performance improvements in cellular phones, laptops, and other battery-operated devices.

Reversible computers, also called *adiabatic systems*, recycle their energy and thus emit very little heat. This lets computing power grow without hitting the technology wall created by high-performance chips releasing large amounts of heat.

Energy efficiency and heat reduction are important for mobile devices, which run on limited-life batteries and have few resources to provide cooling; for desktops, which are becoming more powerful but are also running hotter and hotter; and potentially for switches and routers, which are increasingly fast and functional and thus consume more energy and generate more heat.

Adiabatic systems conserve energy because they delete little or no data. Computing systems discharge energy and give off heat when storage nodes suddenly change their voltage from positive to negative, which occurs when a bit changes its value. Thus, deleting information uses energy, and the less information that systems erase, the less power they use.

Moreover, rather than use up new energy sources, reversible systems reuse the energy already in their circuits from holding on to the data they contain.

"A normal computer throws away all that energy," explained Assistant Professor Michael Frank of the University of Florida's College of Engineering, who has developed reversible circuit designs over the past few years and is currently creating a demonstration chipset.

Reversible systems work with algorithms that don't require data to be erased. This includes algorithms that run ongoing processes—such as the addition of a long list of numbers—that build on one another and thus don't need to eliminate early steps.

Frank received a $40,000 grant from the Semiconductor Research Corporation, an industry semiconductor-research-management consortium, to conduct a feasibility study on a proof-of-concept reversible computer.

SRC is interested because reversible computing is one of many possible methods for reducing a system's heat level while allowing chip performance to scale, said Ralph Cavin, the consortium's vice president for research operations.

MIT's Reversible Computing research group developed a proof-of-concept reconfigurable reversible chip called Flattop.

However, MIT senior research scientist Tom Knight said he is no longer working on the project. Knight explained that he found it impractical to maintain both high performance and low levels of heat generation and energy usage with reversible computing.

Reversible computing would require new hardware designs, software, development tools, programming languages, and compilers.

Businesses traditionally have resisted such major system architectural changes but may not have much choice when it comes to reversible computing. Frank projects that by 2030, increased heat production will overwhelm conventional computer systems, although some improvements could extend traditional approaches by another five to 10 years.

He estimated that adiabatic technology may begin showing up in embedded systems possibly during the next few years, in portable devices by about 2007, and in desktops by about 2013. ■

—Linda Dailey Paulson

Editor: Lee Garber, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; l.garber@computer.org

Billion-Transistor Architectures: There and Back Again

Doug Burger
The University of Texas at Austin
James R. Goodman
University of Auckland

A look back at visionary projections made seven years ago by top researchers provides a context for the continuing debate about the future direction of computer architectures.

In September 1997, *Computer* published a special issue on billion-transistor microprocessor architectures.[1] Our goal for that issue was to frame the debate about the direction computer architectures would take as Moore's law continues its relentless drive down to atomic limits. That issue, widely cited, contained a range of visionary projections from many top researchers, covering the space of architectural possibilities at the time across a range of candidate markets and domains.

We solicited and selected two sets of papers. The first set enumerated important emerging trends that were potential drivers of architectural change in technology, applications, and interfaces. The second set described a number of visions for designs that could and would scale up to billion-transistor architectures (BTAs). What emerged from the accepted set of papers was that there was no consensus about which direction microprocessor architectures are likely to take as chip integration reaches unprecedented levels.

Seven years later, it is both interesting and instructive to look back on that debate and the projections made. What did the community get right? What did we miss? What new ideas have since emerged?

It turned out that many of the authors—a surprising number given the disparity in opinions—were exactly right about the directions that industry would take in the near term. However, none of the architectural models discussed has become dominant, and it is still unclear that any of them will be the right model for BTAs across a broad swath of future markets.

LOOKING BACKWARD AT FORWARD PROJECTIONS

Architectures are never designed in a vacuum—they are always affected by technology, cost, customers, workload, and usability constraints, as well as marketing initiatives and fads. Because of the complexity of modern systems, there is also tremendous pressure for architectures to evolve gradually; major transitions are extremely rare. Consequently, it is important to understand the specific constraints that cause evolution in architectures over time.

Architecture-affecting trends

The 1997 issue contained articles that each predicted a driving external force that would affect architectures over the next decade. These constraints fell into three categories—technology, workloads, and hardware/software interfaces.

Published by the IEEE Computer Society

Technology. Doug Matzke[2] presented a prescient study of on-chip interconnect delay, predicting that because of faster clocks and growing resistivity in shrinking wires, only small fractions of a chip would be reachable in a single cycle by 2012. Although many researchers were aware of this trend—several of the articles cited wire delays as a key driving issue—this study quantified the extent of the emerging challenge, unrecognized by many at the time. Matzke's projections still hold: The effects of slower wires continue to increase each year.

Workloads. Keith Diefendorff and Pradeep Dubey[3] made the case that multimedia workloads would be the key driver of new computer architectures. In particular, they predicted that the general-purpose processor market would subsume the high-end DSP market as BTAs inevitably incorporated support for efficient multimedia execution: real-time capabilities, loop-specific optimizations, and subword data parallelism. This unrealized convergence is still possible because graphics and signal processing systems are becoming more programmable, and future general-purpose machines are likely to exploit more fine-grained concurrency. Whether the two types of architectures will converge remains an open question.

Binary interfaces. Josh Fisher[4] made the case that fixed instruction sets would become less important due to "walk-time techniques" such as binary translation and dynamic recompilation, enabling many minor application- or market-specific variations in each family of instruction sets with full software cross-compatibility. This capability has not yet become universal, but individual companies like Transmeta—whose chips run legacy x86 code on a VLIW implementation—rely on such technology. Additionally, there is evidence that the major microprocessor vendors are moving in this direction.

Projections for future BTAs

The 1997 issue included visions of what future BTAs would be from seven top research groups, selected to cover the spectrum of leading candidate architectures. While seven years is a short time in terms of design generations—fewer than two assuming a four-year design cycle—it is simultaneously a long time in our fast-paced field.

We ordered the articles in the 1997 issue according to the granularity of parallelism exposed to software—coarsest to finest—which influences the ease of partitioning the hardware. The spectrum of granularities ranged from a single thread running on a single, enormous, wide-issue superscalar processor to a chip with numerous small, single-issue tiles in which both the computation and the interfile communication are fully exposed to software.

Software and hardware may be headed for a train wreck.

This debate about the correct degree of partitioning is timely because software and hardware may be headed for a train wreck. The increasing wire delays that Matzke described are forcing greater partitioning of hardware, which could in turn force more partitioning of software. Because many applications are still monumentally difficult to parallelize, hardware designers may provide more processing units but pass the buck to either compilers or programmers to figure out how to use them. The right point in this space (for each application class) must carefully balance this tension between hardware and software partitioning.

Wide-issue superscalar processors. Yale Patt and his group[5] advocated ultrawide-issue, out-of-order superscalar processors as the best alternative for BTAs. They predicted that the first BTAs will contain a single 16- or 32-wide-issue processing core using out-of-order fetch, large trace caches, and huge branch predictors to sustain good instruction-level parallelism (ILP).

At present, industry is not moving toward the wide-issue superscalar model; the termination of the Alpha 21464 design—an 8-wide-issue, multithreaded out-of-order core—was a significant setback. This model suffers from high design complexity and low power efficiency, which are both currently of enormous concern to product groups. Since these issues have not been mitigated, industry is moving in other directions: The desktop market has continued with narrow-issue, ultra-high-frequency cores; the server market has begun using multithreaded chip multiprocessors; and the graphics market is starting to use CMPs that are more fine-grained than server processors. New types of instruction-set architectures may move wide-issue superscalar processors back into favor.

Superspeculative superscalar processors. Mikko Lipasti and John Shen[6] proposed Superflow, a wide-issue superscalar architecture that relied on heavy data speculation to achieve high performance. Like Patt's group, they assumed an aggressive front end that used a trace, but differed by proposing a data speculation engine that used value prediction for loads, load addresses, and arithmetic instructions, along with load/store dependence prediction for memory ordering.

Aggressive speculation has become commonplace throughout microprocessor pipelines, but it

Designers need to find the sweet spot between single-thread execution semantics and a distributed architecture.

has not yet broadly incorporated value speculation. Most modern predictors mitigate performance losses due to deeper pipelines; as industry has progressively shortened the clock period, state previously reachable from a given point becomes unreachable in a single cycle, forcing the microarchitecture either to wait or to guess. Thus, of the speculative techniques that Lipasti and Shen advocated, those that facilitated deeper pipelines have generally been implemented, but most of the techniques intended to support high ILP in a wide-issue machine have not.

Simultaneous multithreaded processors. SMT processors share a superscalar core dynamically and concurrently, increasing its utilization. Susan Eggers and coauthors[7] accurately predicted that SMT processors would appear in the near future—both the Intel Pentium 4 and IBM's Power5 processor use SMT technology. However, the number of threads per individual core is unlikely to increase much beyond the small number currently appearing, making SMT an unlikely first-order paradigm for BTAs. All superscalar-style cores likely will have some form of SMT capability, but SMT is not a model that will provide long-term scalability for future implementations.

Distributed processors. James E. Smith and Sriram Vajapeyam[8] advocated trace processors as a viable candidate for BTAs. They argued that logical uniprocessors—running a single thread—are desirable, but because hardware trends will increase the necessary partitioning, microarchitectures will inevitably start to resemble parallel processors. They described trace processors as an example of a "fourth-generation architecture" in which a single logical thread feeds multiple discrete processing engines, one trace at a time. Trace processors are one approach to finding the sweet spot between single-thread execution semantics and a necessarily distributed microarchitecture.

Aside from limited clustering in the Alpha 21264, designers have not yet adopted aggressive microarchitectural partitioning, although recent academic literature frequently describes clustered microarchitectures. To tolerate wire delays, high-frequency processor designers have instead added pipeline stages for communication—for example, the Pentium 4—rather than clustering the execution core. Adding pipeline stages is a short-term solution for wire delays, so clustering is inevitable for large processors that support single threads.

Vector IRAM processors. Christoforos Kozyrakis and colleagues[9] advocated placing enormous, high-bandwidth memories on the processor die—built using dynamic RAM (DRAM) technology—integrating physical memory with the processor and thus increasing main memory bandwidth appreciably. They proposed using vector processors to exploit this additional bandwidth and developing new compiler techniques to vectorize many applications previously deemed unvectorizable.

The importance of vector-like media processing has clearly increased, and vector processors have remained important at the ultrahigh end of the computing spectrum—for example, the Japanese Earth simulator. However, the continued divergence of DRAM and logic processes makes vector intelligent RAM (VIRAM)-like parts unlikely to subsume general-purpose processors anytime soon. Vector-like processors with dedicated and integrated memories are good candidates for data-parallel workloads in the embedded space.

Chip multiprocessors. Like many of the other authors, Lance Hammond and coauthors[10] argued that wire delays and changing workloads will force a shift to distributed hardware, which in their model consists of a large number of simple processors on each chip. Unlike other authors, they extended that argument to software, claiming that the programming model is likely to change to exploit explicit parallelism because a CMP uses transistors more efficiently than a superscalar processor only when parallel tasks are available.

In the high-performance commercial sphere, CMPs are becoming ubiquitous. IBM's Power4 has two processors, Compaq WRL's proposed Piranha processor had eight, and Intel has announced plans to build CMP-based IA-64 processors. In the desktop space, however, single-chip uniprocessors are currently still dominant. A key question is whether CMPs—made up of simple processors—can scale effectively to large numbers of processors for nonserver workloads. Computer architecture historians may be interested to know that the 1997 *Computer* issue was where the now widely used CMP acronym was popularized, although we had first used the term a few months before in a paper presented at ISCA.

Raw microprocessors. Finally, Elliot Waingold and coauthors[11] proposed Raw microprocessors as the right model for BTAs. These processors have the flavor of a highly clustered, two-dimensional VLIW processor in which all of the clusters have independent sequencers. Raw processors push partitioning to an extreme, with numerous extremely simple and highly distributed processing tiles managed wholly by software. Statically scheduled instruction streams at each intertile router manage interprocessor communication.

These systems achieve terrific scalability and efficiency for codes exhibiting statically discoverable concurrency, such as regular signal processing applications. However, they still cannot deal effectively with runtime ambiguity, such as statically unpredictable cache misses or dynamically determined control, making them unlikely candidates for BTAs except in specialized domains.

EMERGING TRENDS

A number of constraints and trends, the significance of which many researchers (including us) did not foresee, have emerged since 1997. Some of these new directions are affecting the march toward balanced and scalable BTAs.

Superclocked processors

The extent to which faster clocks were driving designs was known but underappreciated seven years ago. Since then, industry has continued along the high-frequency path, emphasizing faster clock rates over most other factors. This emphasis is most clearly evident in the Intel x86 family of processors.

In 1989, Intel released the 80386, implemented in approximately 1-µm technology, with a 33-MHz clock rate. That frequency corresponded roughly to 80 fan-out-of-four (FO4) inverters' worth of logic per clock cycle, with each inverter driving a load four times that of its own. By 2003, Intel was selling 3.2-GHz Pentium 4 chips, implemented in roughly 90-nm (or .09-µm) technology—a 100-fold increase in frequency. This speed increase came from two sources: smaller, faster transistors and deeper pipelines that chopped the logic up into smaller pieces. The Pentium 4 has between 12 and 16 FO4 inverters per clock cycle, a decrease of 80 to 85 percent compared to the 80386.

This rapid clock speed increase—40 percent per year over the past 15 years—has provided most of the performance gains as well as being the primary driver of microarchitectural changes, a result that few researchers predicted. Most of the new structures and predictors appearing in complex microarchitectures, such as load latency predictors in the Alpha 21264 and the Pentium 4, are there solely to support high frequencies, mitigating the ILP losses resulting from deeper pipelines.

The emphasis on frequency increases has had three major implications. First, it has hastened the emergence of power bottlenecks. Second, it has deferred the need to change instruction sets; since RISC instruction sets, and the x86 µop equivalents, were intended to support pipelining effectively, industry was able to focus on clock scaling without incurring the pain of changing industry-standard architectures. The dearth of new ISAs in the past 15 years is more attributable to the explosion of clock frequency than to a fundamental end of ISA innovations. Once design-enabled frequency improvements are no longer viable, we are likely to see a resurgence of ISA changes, although they will likely be hidden behind a virtual machine with an x86 interface.

Third, reductions in the logic-per-clock period are nearing a hard limit; prior work has shown that reducing the clock period much below 10 FO4 inverters per cycle is undesirable.[12,13] We are thus quite close to a microarchitectural bound on frequency improvement. Further, leakage power is likely to bound the rate of device-driven frequency improvement. These two factors suggest that the rate of frequency increases is about to slow dramatically, forcing a shift to other strategies for achieving performance.

The rate of frequency increases is about to slow dramatically, forcing a shift to other strategies for achieving performance.

Power

One factor that has become drastically more important than any of the 1997 authors predicted is power consumption, both dynamic and static. Power issues have moved from being a factor that designers must simply consider to become a first-order design constraint in future processors.

The primary cause of the sudden emergence of dynamic power as a constraint is the extraordinarily rapid and continued growth in clock speeds. Future BTA designs must consider power efficiency as a factor in determining the right way to extract performance from a given software workload—a necessity that penalizes the conventional wide-issue superscalar approach.

Static power is just beginning to emerge as a serious design constraint, but it could be more fundamental by limiting the number of devices available for use at any given time.

Intel's recent announcement of new materials—presumably improved dielectrics—offers some hope that leakage will not limit available devices as soon as some thought. However, we could still eventually find ourselves in a domain in which transistors continue to shrink but do not get faster, putting more pressure on extraction of concurrency for performance rather than raw clock speed.

These potential new power constraints imply that designers must balance high performance with efficient use of transistors, adding another new constraint—wire delays being the other—to options for BTAs.

Researchers are actively exploring two directions: making processors *faster* and making them *better*.

LOOKING FORWARD AGAIN: SOME NEW DIRECTIONS

Semiconductor process experts predict a continued increase in transistor counts for at least another decade. These increases will enable an enormous degree of integration, but the pressing question is, what should we do with all of this hardware? To answer this question, researchers are actively exploring two directions: making processors *faster*, which was the focus of the 1997 articles, and making them *better*.

Making systems better, not faster

As on-chip devices become extraordinarily small and more numerous, using them intrinsically becomes more difficult. They are less reliable, fail more often, and can consume too much power. Furthermore, programmers, languages, and compilers may not be able to use them all effectively. Numerous ongoing research efforts are addressing these challenges by allocating a fraction of future hardware budgets to mitigate the downsides of such enormous device counts.

Assist threads. Since enough explicit parallel threads often are not available, researchers have begun using the parallel thread slots available in SMT processors for "helper" threads. These helper threads are designed to improve performance and have been called variously subordinate threads, slipstreaming, speculative data-driven threads, or master-slave threads.[14-17] Like SMT, this approach could benefit a few generations of designs, but it is not a substitute for scalable hardware or more effective parallel programming.

Reliability, debugging, and security. David Patterson has recently been making the case that reliability in future systems will be paramount and should be more of a focus for researchers than improved performance. Certainly, many recent reports in the literature have focused on providing reliable execution, whether with a result checker,[18] reliability-enhancing redundant threads,[19,20] or a system that supports execution near the edge of tolerable voltage limits.[21]

Researchers have also begun using threads to support software debugging. In related efforts, they have proposed using hardware support to enhance security, for example, detecting and preventing buffer overflows and stack smashing, or providing fine-grained memory protection.[22] Detecting bugs, recovering from faults, and foiling intruders (malevolent and otherwise) are all likely to be important uses for future hardware resources.

Parallel programming productivity. A major underlying theme that emerged from the articles in the 1997 issue was the tension between the difficulty of explicitly partitioning software and the need to partition future hardware. It is clear that the ability of software—either compilers or programmers—to discover concurrency will have a first-order effect on the direction of BTAs in each market. If parallel programming remains intractably difficult for many applications, chips with small numbers of wide-issue processors will dominate, bounded only by complexity and efficiency limits.

We (Jim Goodman, along with his colleague, Ravi Rajwar) have been developing hardware support that improves the ease of productive parallel programming by enabling concurrent execution of transactions.[23] Speculative Lock Elision allows programmers to include locks that suffer no performance penalty if no lock contention occurs, and the more aggressive Transactional Lock Removal[24] provides lock-free execution of critical sections. Programmers can thus concentrate on getting the synchronization code right, with a generous use of locks less likely to kill a program's performance.

Continuing the quest for performance

As frequency improvements diminish, increased concurrency must become the primary source of improved performance. The key concern that architects must address is the number and size of processors on future CMP chips. Scaling the number of simple processors in a CMP beyond a few tens simply doesn't make sense given the state of software parallelization, and it will result in asymptotically diminishing returns. Similarly, scaling a single core to billions of transistors will also be highly inefficient, given the ILP limits in single threads. In our view, future BTAs should have small numbers of cores that are each as large as efficiently possible.

The sizes and capabilities of these large future processors are an open question. The Imagine processor[25] and the follow-on streaming supercomputer effort[26] both use large numbers of arithmetic logic units to exploit the data-level parallelism prevalent in steaming and vector codes, with high power efficiency per operation. We (Doug Burger, along with his colleague, Steve Keckler) have proposed an alternative approach that exploits concurrency from irregular codes and from individual threads using large, coarse-grained processing cores. These large cores rely on a new class of dataflow-like instructions sets called EDGE architectures (for explicit data graph execution—a term that Chuck Moore coined

while he was at the University of Texas at Austin), of which the TRIPS architecture will be the first instance.[27] By enabling much larger cores to exploit concurrency both within and across threads (and vectors), the hope is that this class of architectures will permit future BTAs to continue effective performance scaling while avoiding the need to build "CMPPs" (chip massively parallel processors).

Future BTAs will be judged by how efficiently they support distributed hardware without placing intractable demands on programmers. This balance must also factor in efficiency; hardware that matches the available concurrency's granularity provides the best power and performance efficiency. Researchers will doubtless continue to propose new models as they seek to find the right balance among partitioning, complexity, and efficiency. Whether the right model for general-purpose BTAs ends up being one of those advocated in 1997, a more recent one such as some of those described in this article, or one that has not yet been discovered, the future for interesting architectures has never been more open.

What is even more exciting—or scary, depending on the reader's perspective—is that the solution to these problems could have fundamental implications for both the software stack and software developers. When efficient, transparent solutions to hardware partitioning reach their scalability limit, hardware designers must pass the buck to software, placing the onus for more performance on the programming model.

The next decade in both architecture and software systems research promises to be even more interesting than the last. ■

References

1. D. Burger and J.R. Goodman, "Billion-Transistor Architectures," *Computer*, Sept. 1997, pp. 46-48.
2. D. Matzke, "Will Physical Scalability Sabotage Performance Gains?" *Computer*, Sept. 1997, pp. 37-39.
3. K. Diefendorff and P.K. Dubey, "How Multimedia Workloads Will Change Processor Design," *Computer*, Sept. 1997, pp. 43-45.
4. J.A. Fisher, "Walk-Time Techniques: Catalyst for Architectural Change," *Computer*, Sept. 1997, pp. 40-42.
5. Y.N. Patt et al., "One Billion Transistors, One Uniprocessor, One Chip," *Computer*, Sept. 1997, pp. 51-58.
6. M.H. Lipasti and J.P. Shen, "Superspeculative Microarchitecture for Beyond AD 2000," *Computer*, Sept. 1997, pp. 59-66.
7. S.J. Eggers et al., "Simultaneous Multithreading: A Platform for Next-Generation Processors," *Computer*, Sept. 1997, p. 49.
8. J.E. Smith and S. Vajapeyam, "Trace Processors: Moving to Fourth-Generation Microarchitectures," *Computer*, Sept. 1997, pp. 68-74.
9. C. Kozyrakis et al., "Scalable Processors in the Billion-Transistor Era: IRAM," *Computer*, Sept. 1997, pp. 75-78.
10. L. Hammond, B.A. Nayfeh, and K. Olukotun, "A Single-Chip Multiprocessor," *Computer*, Sept. 1997, pp. 79-85.
11. E. Waingold et al., "Baring It All to Software: Raw Machines," *Computer*, Sept. 1997, pp. 86-93.
12. A. Hartstein and T.R. Puzak, "The Optimum Pipeline Depth for a Microprocessor," *Proc. 29th Int'l Symp. Computer Architecture*, IEEE CS Press, 2002, pp. 7-13.
13. M.S. Hrishikesh et al., "The Optimal Logic Depth Per Pipeline Stage Is 6 to 8 FO4 Inverter Delays," *Proc. 29th Int'l Symp. Computer Architecture*, IEEE CS Press, 2002, pp. 14-24.
14. R.S. Chappell et al., "Simultaneous Subordinate Microthreading (SSMT)," *Proc. 26th Int'l Symp. Computer Architecture*, IEEE CS Press, 1999, pp. 186-195.
15. Z. Purser, K. Sundaramoorthy, and E. Rotenberg, "A Study of Slipsteam Processors," *Proc. 33rd Int'l Symp. Microarchitecture*, IEEE CS Press, 2000, pp. 269-280.
16. A. Roth and G.S. Sohi, "Speculative Data-Driven Multithreading," *Proc. 7th Int'l Symp. High-Performance Computer Architecture*, IEEE CS Press, 2001, pp. 37-48.
17. C. Zilles and G.S. Sohi, "Master/Slave Speculative Parallelization," *Proc. 35th Int'l Symp. Microarchitecture*, IEEE CS Press, 2002, pp. 85-96.
18. T.M. Austin, "Diva: A Reliable Substrate for Deep Submicron Microarchitecture Design," *Proc. 32nd Int'l Symp. Microarchitecture*, IEEE CS Press, 1999, pp. 196-207.
19. S.K. Reinhardt and S.S. Mukherjee, "Transient Fault Detection via Simultaneous Multithreading," *Proc. 27th Int'l Symp. Computer Architecture*, IEEE CS Press, 2000, pp. 25-36.
20. E. Rotenberg, "AR-SMT: A Microarchitectural Approach to Fault Tolerance in Microprocessors," *Proc. 29th Int'l Symp. Fault-Tolerant Computing*, IEEE CS Press, 1999, pp. 84-91.
21. D. Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *Proc. 36th Int'l Symp. Microarchitecture*, IEEE CS Press, 2003, pp. 7-18.

22. E. Witchel, J. Cates, and K. Asanovic, "Mondrian Memory Protection," *Proc. 10th Int'l Symp. Architectural Support for Programming Languages and Operating Systems*, ACM Press, 2002, pp. 304-315.
23. R. Rajwar and J.R. Goodman, "Speculative Lock Elision: Enabling Highly Concurrent Multithreaded Execution," *Proc. 34th Int'l Symp. Microarchitecture*, IEEE CS Press, 2001, pp. 294-305.
24. R. Rajwar and J.R. Goodman, "Transactional Lock-Free Execution of Lock-Based Programs," *Proc. 10th Int'l Symp. Architectural Support for Programming Languages and Operating Systems*, ACM Press, 2002, pp. 5-16.
25. S. Rixner et al., "A Bandwidth-Efficient Architecture for Media Processing," *Proc. 31st Int'l Symp. Microarchitecture*, IEEE CS Press, 1998, pp. 3-13.
26. W.J. Dally, P. Hanrahan, and R. Fedkiw, *A Streaming Supercomputer*, white paper, Computer Systems Laboratory, Stanford Univ., 2001.
27. K. Sankaralingam et al., "Exploiting ILP, TLP, and DLP with the Polymorphous TRIPS Architecture," *Proc. 30th Int'l Symp. Computer Architecture*, IEEE CS Press, 2003, pp. 422-433.

Doug Burger *is an assistant professor in the Department of Computer Sciences at the University of Texas at Austin. His research interests are computer architecture, advanced compilation, and high-performance computing. Burger received a PhD in computer sciences from the University of Wisconsin-Madison. He is a member of the ACM, a senior member of the IEEE, and a Sloan Foundation Research Fellow. Contact him at dburger@cs.utexas.edu.*

James R. Goodman *is a professor in the Department of Computer Science at the University of Auckland. His research interests focus on computer architecture. Goodman received a PhD in electrical engineering and computer science from the University of California, Berkeley. Contact him at goodman@cs.auckland.ac.nz.*

IEEE is VALUE.

Beth depends on IEEE publications to bring her organization the latest technology research.

Not only do IEEE publications provide Beth with over 1 million online documents – including top-cited journals, cutting-edge conferences and vital standards – they do it for a reasonable price.

Research shows that IEEE journals are just 58% of the average journal price in the field on a page-for-page basis. That's why 81% of IEEE customers are satisfied or highly satisfied with the value of IEEE content. And why Beth has the reputation for spending her budget so wisely.

To Beth, IEEE is value. Discover what IEEE can be for you.

Go here.

www.ieee.org/discover

IEEE

IEEE INFORMATION DRIVING INNOVATION

Metamorphic Programming: Unconventional High Performance

Metamorphic programming solves common computing problems with sometimes spectacular performance gains over conventional coding. Although these programs violate good programming rules, a few minor compiler enhancements can produce clean, well-structured code.

Peter M. Maurer
Baylor University

Five years ago, I would have laughed if I had read an article that stated, "Metamorphic programming can make your code run five to 60 times faster and is so simple that it will transform your programs into straight-line code. The only catch is, you will have to violate most of the rules of good programming, and you may have to invent one or two new algorithms."

Now after my students and I have written and tested dozens of metamorphic programs, I'm ready to take the idea seriously. In fact, I've become something of a fanatic.

The catch is real, of course. At the start of our metamorphic exploration, we rapidly discovered it was pointless to merely translate conventional code into a metamorphic counterpart. Instead, given that metamorphic programming's strength is its ability to process a collection of heterogeneous objects, we decided that it made more sense to rewrite existing algorithms and create new ones to exploit it. After a while, we were almost eager to find new applications for it.

The speed and performance gains we achieved in simulation were nothing short of spectacular. For a new VLSI circuit, faster simulation translates to faster time to market, so even the most peculiar programming type is worth exploring if the benefit is increased performance. When we ran the first metamorphic programming experiments to reduce simulation time, we would have been happy with a 50 percent increase. What we got was anywhere from 500 to 700 percent, as the "A Simulation Experiment" sidebar shows.

As we continued to use this technique in logic-level VLSI circuit simulation, we found efficient and elegant solutions for many programming problems. The resulting performance gains inspired us to look for metamorphic solutions to problems outside simulation.

We have since implemented metamorphic solutions for many common computer science problems and are convinced that metamorphosis could be a powerful tool for any algorithm that uses state data. String matching and lexical analysis, for example, are explicitly state based and thus readily adaptable to metamorphic techniques. Graph algorithms that maintain state data, such as shortest-path and depth-first search, are also good candidates. Even straightforward algorithms like sorting are somewhat state based, since the algorithm's behavior changes at the end of a list.

Space dictates that I offer only a sampling of our metamorphic programming applications, but the

 Published by the IEEE Computer Society 0018-9162/04/$20.00 © 2004 IEEE

full account is posted at http://cs.ecs.baylor.edu/~maurer/Metamorphic.

METAMORPHOSIS IN A NUTSHELL

As its name implies, metamorphic programming is about handling change, which is especially useful in an object-oriented (OO) environment. Objects change their identity during program execution, functions are redefined, and even data items can change type or become hidden. Object behavior changes over time as the object adapts to differing conditions or responds to new needs. Metamorphic programming is a good fit for object states, particularly those that affect object behavior.

Existing OO algorithms handle state information by using state variables that operations decode to produce the required behavior for a particular state. This decoding is inefficient, however, because it duplicates work.

To illustrate, consider a binary semaphore *S* with two states, 1 and 0. The *P* and *V* operations do two distinctly different things, depending on the state of *S*. In state 1, the system changes the *P* operation to state 0 and ignores the *V* operation. In state 0, the *P* operation blocks the calling process, while the *V* operation either unblocks a process or changes back to state 1. If you initialize the semaphore to state 1 and perform the *V* operation, the state will change from 1 to 0. At the moment of change, you know the *P* and *V* functions' new behavior, but a typical implementation discards this knowledge, encoding the state to 0 or 1. The generic *P* and *V* functions must then decode the state to determine the correct behavior. The job of these operations is thus to recover information that was readily available earlier—not very efficient.

A more elegant strategy is to have a separate set of *P* and *V* functions for each state, which eliminates the need to test the state or even to record its value. Figure 1 shows the code with pointers to the new functions, which the program uses to complete runtime binding. This isn't exactly legal C++, but you get the idea.

Replacing the *P* and *V* functions changes the semaphore's behavior and effectively changes its identity. In state 0, the semaphore is an object that does nothing. In state 1, it is an object that queues and dequeues processes.

EXTENDING POLYMORPHISM

For all its apparent strangeness, metamorphosis has many characteristics of the polymorphic types in conventional OO programming. Polymorphism—types created using inheritance and virtual functions—processes a heterogeneous set of objects without using type codes or type decoding.[1,2] Metamorphosis extends that to dynamic codes. Coincidentally, the mechanisms to implement the two are similar.

Key to polymorphism are virtual functions, which, unlike conventional functions, are bound to their function calls at runtime. The class definitions in Figure 2 illustrate this idea. The pointer variable, MyPtr, can point to an object of type MyPoly or an

A Simulation Experiment

Table A compares the speed of a conventional simulator with our metamorphic program (Event-Driven, Conditional Free) using several standard benchmarks.[1] The programs ran on a Sun 300-MHz single processor Ultra Sparc-II with 128 Mbytes of RAM. We used 50,000 random input vectors for each test.

Table A. Speed of a conventional and a metamorphic simulation.

Circuit	Conventional Event-Driven (CPU sec.)	Event-Driven, Conditional Free (CPU sec.)	Speedup
C432	10.8	1.4	7.71
C499	12.1	1.7	7.11
C880	20.2	4.0	5.05
C1355	43.2	5.6	7.71
C1908	82.5	8.1	10.19
C2670	89.3	13.6	6.57
C3540	128.5	15.3	8.40
C5315	252.9	27.5	9.20
C6288	2,549.5	42.1	60.56
C7552	396.8	40.2	9.87

Reference

1. P. Brglez, P. Pownall, and R. Hum, "Accelerated ATPG and Fault Grading via Testability Analysis," *Proc. IEEE Int'l Symp. Circuits and Systems*, IEEE CS Press, 1985, pp. 695-698.


```
P0()                    P1()
{                       {
   P = &P1;                Block Current Process;
   V = &V1;                Queue Current Process;
}                       }

V0()                    V1()
{                       {
   return                  if (Process is queued)
}                             Dequeue & Unblock Process;
                           else {
                              P = &P0;
                              V = &V0;
                           }
                        }
```

Figure 1. Pointers to state-specific* P *and* V *functions for a binary semaphore.

```
class MyPoly
{
   MyPoly * Next;
   virtual void MyFunc( )
   {
      cout<<"Apple";
   }
}
```

```
class MyDerv : public MyPoly
{
   virtual void MyFunc( )
   {
      cout<<"Orange";
   }
}
```

```
MyPoly * MyPtr;
MyPoly * MyPtr;
MyPoly Obj1;
MyDerv Obj2;

MyPtr = &Obj1;
MyPtr->MyFunc( );
MyPtr = &Obj2;
MyPtr->MyFunc( );
```

Figure 2. An example of runtime binding. The executable code assigns a pointer of each type to MyPtr, and then calls MyFunc.

```
Shape * Head;
…
float Total = 0.0;
for (MyPoly * Temp = Head ; Temp != NULL ; Temp=Temp->Next)
{
   Temp->MyFunc();
}
```

Figure 3. Runtime binding to print the type of each object in a list.

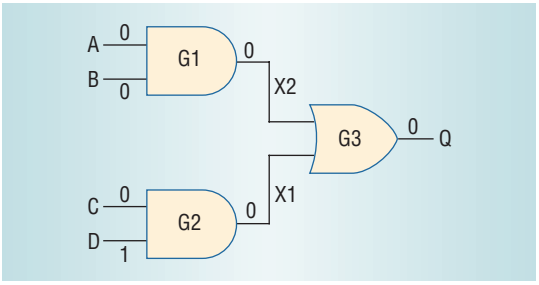


Figure 4. Typical circuit used in simulation experiments.

object of type MyDerv. The executable code assigns a pointer of each type to MyPtr, and then calls MyFunc. Because the program binds MyFunc at runtime, the two calls produce different results: The first prints "Apple;" the second, "Orange." Conventional functions are bound to function calls at compile time, which in this case would cause both function calls to print "Apple."

OO programmers often use runtime binding to process a heterogeneous object collection, as in Figure 3, where the aim is to print the type of each object in a list.

Before polymorphism, programmers used a type code to distinguish between MyPoly and MyDerv. The loop then decoded the type code to determine the correct MyFunc function. Like a state code, the type code represents lost information. When you create MyPolys and MyDervs, you know the correct procedure for printing the message, which means you can append the correct MyFunc function to an object when you create it.

Both polymorphism and metamorphic programming provide opportunities to replace explicit code with subroutine addresses, and, because these addresses give specific behavior, they are significantly more useful than numeric codes.

METAMORPHOSIS AND SIMULATION

In any simulation, the first step is to translate the system or circuit into an object collection. In a typical simulation experiment, a simulator translates a circuit, such as that in Figure 4, into an interconnected collection of gates and nets (wires) with an object for each. Net objects have a value element that maintains the circuit's state. Except for flip-flops, gates seem to have no state, so most simulators treat them as pure functions (retaining the word "gate" for simplicity) with special scheduling techniques to simulate gate and net delays.[3]

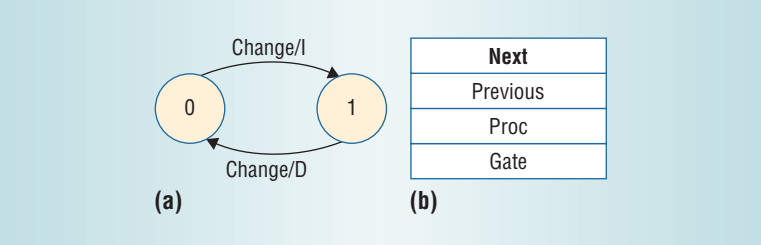


Figure 5. State machine (a) and data structure to implement it (b) for a net in the simulation of the circuit in Figure 4.

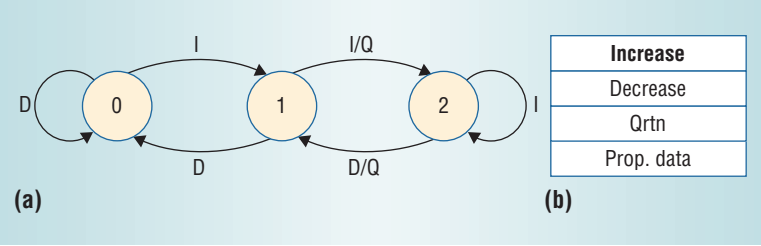


Figure 6. State machine for a two-input AND gate (a) and the data structure to implement it (b).

Our simulator links nets that change value into a queue of pending changes. It inserts objects into the queue's tail and processes them when they reach the queue's head. Each object has one or more functions that change to reflect the object's state. The simulator traverses the linked list and executes each object's current function.

Our dramatic performance increases stem primarily from our decision to model both gates and nets as state machines. Obviously nets have states, since they must have a value of either 0 or 1, but gates also have states. In Figure 4, if either of gate 1's (G1) inputs changes, nothing happens. However, if input C of gate 2 (G2) changes, output X1 changes, and the change propagates through gate 3 (G3) to output Q. Clearly, G1 and G2 are in different states. Nets A and B are then state machines that transmit values to G1, and G1 in turn is a state machine that transmits values to X2.

The states of X1 and X2 are important only because of how they affect G3's state. These nets do not require an explicit 1 or 0 value, so any convenient method for representing state will do. Nets A, B, C, D, and Q needed a 1 or 0 state code because we must examine inputs for changes and report output values to the user.

Figure 5 shows the state machine for a net and the data structure of its implementation. The state machine's input signal is a changed net value. The net's state machine sends output signals *I* and *D* (also known as Increment and Decrement) to the gate's state machine. The data structure's Proc element points to the subroutine that will process the net's next change. One subroutine is for 0-to-1 changes, and another is for 1-to-0 changes. This pointer is the only state information maintained for the net. The Next and Previous elements are for queuing. The Gate element points to the gate that will receive the net's output signals.

Figure 6 shows the state machine for a two-input AND gate and its associated data structure. This machine receives *I* and *D* signals from two nets' state machines. Because the two inputs are symmetric, we did not have to distinguish them, but we did have to track the inputs that equaled 1. If both inputs are 1, the output is 1; otherwise the output is 0. The output changes when there is a transition between states 1 and 2. The resulting *Q* output causes the simulator to add the gate's output to the end of the simulation queue.

As Figure 6b shows, the Incr and Decr elements maintain the machine's state. These elements point to the subroutines that handle the *I* and *D* inputs from the input machines. The input state machines transmit their inputs by calling one of these routines directly. The Qrtn element maintains the gate's queuing state. The queuing actions depend on the timing model.

The code to support the state machines for the nets and gates is surprisingly simple. In the code for the net's state machine, which Figure 7 shows, two subroutines, DProcessor and IProcessor, toggle to maintain the net's state, with the IProcessor calling the Incr subroutine, and the DProcessor calling the Decr subroutine. We replaced tail recursions with computed goto statements to improve performance.

The code for the gate's state machine, in Figure 8, is only slightly more complicated. The machine never calls Decr0 and Incr2 routines. The other four routines change state by assigning new subroutine addresses to Incr and Decr. The Incr1 and Decr2 routines call the queuing subroutine to queue the output

```
IProcessor:
   Cev->Proc = &&DProcessor;
   Cgt = Cev->Gate;
   goto * Cgt->Incr;
```

```
DProcessor:
   Cev->Proc = &&IProcessor;
   Cgt = Cev->Gate;
   goto * Cgt->Decr;
```

Figure 7. Code to support the net's state machine.

```
Incr0:
   Cgt->Incr = &&Incr1;
   Cgt->Decr = &&Decr1;
   Cev = Cev->Next;
   goto *Cev->Proc;

Decr0:
   Cev = Cev->Next;
   Goto *Cev->Proc;

Incr1:
   Cgt->Incr = &&Incr2;
   Cgt->Decr = &&Decr2;
   goto *Cgt->Queue;

Decr1:
   Cgt->Incr = &&Incr0;
   Cgt->Decr = &&Decr0;
   Cev = Cev->Next;
   Goto *Cev->Proc;

Incr2:
   Cev = Cev->Next;
   goto *Cev->Proc;

Decr2:
   Cgt->Incr = &&Incr1;
   Cgt->Decr = &&Decr1;
   Goto *Cgt->Queue;
```

Figure 8. Code to support the gate's state machine.

```
for (long i=1 ; i<n ; i++)
{
   long x = L[i];
   for (long j=i-1 ; j>=0 && L[j]>x ; j--)
   {
      L[j+1] = L[j];
   }
   L[j+1] = x;
}
```

Figure 9. Iterative algorithm for sorting objects in a list.

net, while the Incr0 and Decr1 routines advance to the next queued net. As before, we used computed goto statements instead of subroutine calls.

The rest of our simulator's code is similar to that in Figures 7 and 8.[3-6] The subroutines contain assignment statements, but no conditional statements and no loops, and we use computed goto statements instead of subroutine calls. The code is a straight-line series of assignments with a few labels and computed goto's.

At best, our simulation algorithm looks peculiar, and at worst it violates most good coding rules. Far from considering the goto harmful, we see it as our most important tool—and *computed* goto's at that. We also don't believe in static object definitions. Indeed, we have gone to extraordinary lengths to violate this rule, even to the extent of inserting assembly language into high-level programs—hardly the sort of thing Computer Science 101 would recommend.

Admittedly, our code is unconventional, but what else could we have done with the available tools? Function pointers don't offer much support because we can't restrict object metamorphosis to a specified collection of definitions. We could have used the State pattern from the gang-of-four patterns,[7] but that would have been no more elegant or efficient than our solution.

Goto's are the only resort when tools give you no way to specify cheap function calls. When you go from one subroutine to another, you don't need a new stack frame, a new return address, new parameters, and new local variables. You just need to *get* from one place to another. Why pay for a bunch of stuff you don't need? If you feel lucky, you could use tail recursion and ordinary function calls and then cross your fingers that the optimizer would undo all the damage. However, that is rather like leaving off the inline keyword and hoping the compiler will guess what needs to be expanded in-line.[8]

So, no, we don't need better code. We just need better tools.

MORE ALGORITHMS

Of course, one algorithm is hardly the basis for launching tool development, so it is worth mentioning a few more of the metamorphic algorithms we created before talking about tools.

Insertion sort

Our insertion-sort algorithm is based on the iterative algorithm in Figure 9, which sorts objects in a doubly linked list by calling the same function for each object.

Each object in the sort has two functions: a forward routine and a backward routine. The forward routine replaces the iterative algorithm's outer loop; the backward routine replaces its inner loop. Figure 10 gives the code for the forward and backward routines. Each object to be sorted points to the forward and backward functions. During the forward traversal, the algorithm removes each object from the list and reinserts it into its proper position in the sorted portion of the list. A backward traversal locates the proper position for the removed element. After it reinserts the removed element, the algorithm resumes the forward traversal. Terminator objects pointing to the EOL and SOL routines are at the ends of the list to terminate traversals.

This example illustrates one of the most important benefits of metamorphic programming: the elimination of "Are we there yet?" programming. The iterative insertion-sort algorithm is like a child on a long trip who continually asks "Are we there yet?" The outer loop executes the same test over and over, searching for the end of the list. In OO programming, objects should know when they're at the end of the list. Repetitive testing shouldn't be required. Admittedly we've cheated a bit by using terminator objects, but rewriting the algorithm could easily eliminate them.

```
Forward:
   This = Current;
   Current = Current->Next;
   BackPtr = This->Prev;
   // unlink;
   This->Next->Prev = This->Prev;
   This->Prev->Next = This->Next;
   goto * BackPtr->BackwardRtn;
```

```
Backward:
   if (This->Value < BackPtr->Value)
   {
      BackPtr = BackPtr->Prev;
      goto * BackPtr->BackwardRtn;
   }
   else
   {
      This->Next = BackPtr->Next;
      This->Prev = BackPtr;
      BackPtr->Next->Prev = This;
      BackPtr->Next = This;
      goto * Current->ForwardRtn;
   }
```

```
EOL:
   return
```

```
SOL:
   This->Next = BackPtr->Next;
   This->Prev = BackPtr;
   BackPtr->Next->Prev = This;
   BackPtr->Next = This;
   goto * Current->ForwardRtn;
```

Figure 10. Forward and backward routines in the metamorphic version of the iterative algorithm in Figure 9.

```
Test:
   if (This->Value < Pivot->Value)
   {
      Split++;
      Swap(This->Value,
         Split->Value);
   }
   This++;
   goto * This->Process;
```

```
LastTest:
   if (This->Value < Pivot->Value)
   {
      Split++;
      Swap(This->Value,
         Split->Value);
   }
   Swap(Pivot->Value,
      Split->Value);
   // Demorph last element
   This->Process = &&Test;
   // push first sublist
   List->First = First;
   List->Last = Split-1;
   List->Process = &&NewList;
   List++;
   // iterate through 2nd sublist
   First = Split+1;
   goto NewList;
```

Figure 11. Test and LastTest routines in Quicksort, a metamorphic sorting algorithm.

Quicksort

In metamorphic programming, arrays work just as well as linked lists, so our Quicksort algorithm stores the objects to be sorted in an array. When a list splits, the algorithm creates two sublists. It continues this action iteratively with one list and pushes the other onto a stack. If the current list contains fewer than two elements, the algorithm pops the stack. The popping continues until Quicksort finds a list with two or more elements or until the stack is empty. When the stack becomes empty, the algorithm terminates.

Quicksort splits a list by calling each object's Process function. Each object has two data items: a value and a pointer to a processing routine. The last list element points to the LastTest subroutine. All other objects point to the Test subroutine. Figure 11 gives the code for the Test and LastTest routines. Because lists divide into progressively smaller sublists, the algorithm morphs the last object in each list into a list terminator.

Stack processing is also metamorphic. Each stack element is an object that contains the list boundaries and a pointer to a processing routine. The last stack element is a terminator whose processing routine terminates the sort algorithm.

Figure 12 gives the NewList routine, which sets up a new list and pops the stack.

ADDING TO THE TOOL BOX

If metamorphic programming is ever to become a serious alternative to iterative programming, we

```
NewList:
   if (Last <= First)
   {  // Pop List
      List--;
      First = List->First;
      Last = List->Last;
      goto * List->Process;
   }
   // Set up list and process
   Split = First;
   Pivot = First;
   This = First+1;
   Last->Process = &&LastTest;
   goto * This->Process;
```

Figure 12. NewList routine in Quicksort.

```
class Cexample
{
public:
   void ABC(void) one of A, B, C;
private:
   void A (void)
   {
      …
   }
   void B (void)
   {
      …
   }
   void C (void)
   {
      …
   }
   void D (void)
   {
      …
   }
};
```

Figure 13. Function ABC, which acts as a dynamic reference to class A, B, or C.

need to eliminate its warts. The first step in that process is to create real metamorphic objects, declare them as such, and specify rules that a compiler can check—not just program objects metamorphically.

This task is not as radical as it sounds because someone has implemented almost anything you can think of in some language somewhere, and metamorphosis is no exception. In our research, however, we can't afford to focus on arcane or experimental languages. To build simulation tools that any programmer can use to verify next-generation VLSI circuits, we must look to mainstream languages like Java, C++, and Visual Basic. Fortunately, the changes required to support metamorphosis in these languages are relatively minor.

An object implementation's low level clearly shows the simplicity of metamorphic language features. Consider three classes, A, B, and C, in which class A is the base type from which we derive classes B and C. Classes B and C do not define any new data items. Class A has a number of virtual functions that classes B and C override. Other than the overrides, these two classes define no new functions. For each class, the compiler creates a *vtable*, an object that contains a pointer to each virtual function the class defines or inherits. Each class object contains a pointer to the vtable.

Because of the restrictions we have placed on classes B and C, all four objects have the same data items, and all three vtables have the same layout. By replacing the vtable pointer, you can morph objects between types A, B, and C, which is essentially complete metamorphosis because the new vtable pointer replaces all virtual functions. Complete metamorphosis is quite trivial to implement. Syntactically, you could use a statement such as

```
Morph Object1 [from A] to B,
```

implement it in one or two assembly-language instructions, and easily verify it with few if any changes to the class-definition syntax. (The sample statement is deliberately verbose for clarity. Compiler designers will probably choose something more elegant.)

You could go one step further with complete metamorphosis and permit classes A, B, and C to define different sets of data items. In this case, you would have to formally declare A, B, and C as mutually morphable classes, since any object of type A would need to contain all data items B and C declared, even though A's functions could not access these items.

Although it is trivial, complete metamorphosis is not always feasible, especially if you want to combine several state machines into a single object. For example, suppose a single object embeds five state machines, with five states per machine. To model the object using complete metamorphosis, you would have to define 3,125 classes to capture all state combinations. Partial metamorphosis simplifies the construction of such objects because you can replace individual vtable members but give each object its own vtable copy. You could also integrate the affected vtable portion into the object itself, eliminating the double indirection.

One strategy is to fake partial metamorphosis using function pointers. Problems arise, however, because the function type is the only tool a compiler has for determining a correct assignment to a function pointer. If the program assigns an incorrect value to a pointer variable, diagnosing the resulting errors can be difficult. A better alternative is to

```
class MyObj
{
public:
   MyObj *Next;
   primary Exam1 void ProcessList(void)
   {
      ProcessObject();
   }
   segment Exam1 virtual void ProcessObject(void) = 0;
};

class OtherObj1 : public MyObj
{
public:
   segment Exam1 virtual void ProcessObject(void)
   {
      // process object here
      if (Next != NULL)
      {
         Next->ProcessObject();
      }
   }
};

MyObj * Head;
```

Figure 14. Treating code segments as metamorphic functions. Only when the program calls a function in a different group will it create a new stack frame and return address.

restrict the list of functions that can participate in a metamorphosis operation.

The code in Figure 13 defines a function ABC, which has no body of its own, but will act as a dynamic reference to either A, B, or C. The function headers of ABC—A, B, and C—must be identical. Even though the function header of D is identical to that of ABC, ABC cannot refer to D. As the following statement shows, you can enhance the morph statement for partial metamorphosis. In this case, we made the statement's operands functions instead of objects and classes.

```
morph ExampleObject.ABC to
ExampleObject.A;
```

METAMORPHIC FUNCTIONS

Even though we've committed the worst of all sins by using computed goto's, a slight change in compiler technology will produce clean, well-structured code. Each routine is a segment of a larger function, but don't think of them this way. Programming is easier if you treat the code segments as functions, and the goto's as function calls. Because these two "functions" share the stack frame, you can view the "function call" as replacing the body of the current function. Thus, we tend to think of a code segment as a metamorphic function that dynamically transforms into some other function.

The concept is similar to multithreading. Threads are cheap processes that share the same address space and other resources; metamorphic functions are cheap function calls that share stack frames. You should organize metamorphic functions into mutually morphable groups so that when a function calls another function from its own group, the program will not create a new stack frame. Only when the program calls a function in a different group will it create a new stack frame and return address.

The declaration should be similar to the C++ inline declaration, except that a group name will identify the group to which the function belongs. Figure 14 illustrates this syntax with the keyword *segment*. We also use the keyword *primary* to identify functions that can be called from outside their group.

Although most of our applications clearly distinguish between a function's primary entry point and its internal segments, we are not convinced that this would be a useful distinction in a more general context. A single keyword may suffice for all metamorphic functions.

In the code in Figure 14, we assume that several different classes derive from MyObj, and that each of these overrides the function ProcessObject. The variable Head is the head of an object list. To process this list, we use the single function call

```
Head->ProcessList();+
```

The code requires no loop. We could avoid testing the Next variable for NULL by using a trailer object whose sole function is to terminate a linked object list or by using a different function for the last object in the list.

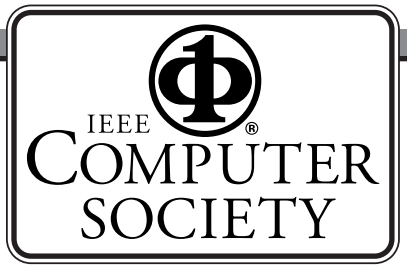
Once we got over the shock of seeing code that violates many of the accepted rules of good programming, we found metamorphic programming to be an effective tool in our search for more efficient algorithms, particularly in logic simulation. Virtually all our implementations have given us some performance increase, although the results from logic simulation are still far more significant. On the other hand, we are just scratching the surface of metamorphic programming. Further experimentation could explore the myriad variant solutions to problems presented here, and they might yield more efficient solutions than ours.

Discovering the most efficient and effective metamorphic techniques across a range of problems will require a concerted effort across the software community. The most important problem is the lack of metamorphic constructs in mainstream high-level languages. My hope is that such features will appear in the near future, and that others will discover the importance of metamorphic programming as a software development tool. ■

References

1. L. Cardelli and P. Wegner, "On Understanding Types, Data Abstraction, and Polymorphism," *ACM Computing Surveys*, Dec. 1985, pp. 471-522.
2. M. Abadi and L. Cardelli, *A Theory of Objects*, Springer, 1996.
3. P. Maurer, "The Shadow Algorithm: A Scheduling Technique for both Compiled and Interpreted Simulation," *IEEE Trans. Computer-Aided Design*, Sept. 1993, pp. 1411-1413.
4. P. Maurer, "The Inversion Algorithm for Digital Simulation," *IEEE Trans. Computer-Aided Design*, July 1997, pp. 762-769.
5. P. Maurer, "Event-Driven Simulation without Loops or Conditionals," *Proc. Int'l Conf. Computer-Aided Design*, IEEE CS Press, 2000, pp. 23-26.
6. D. Lewis, "A Hierarchical Compiled Code Event-Driven Logic Simulator," *IEEE Trans. Computer-Aided Design*, June 1991, pp. 726-737.
7. E. Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
8. G.L. Steele Jr., "Debunking the 'Expensive Procedure Call' Myth, or Procedure Call Implementations Considered Harmful, or Lambda, the Ultimate GOTO," *Proc. ACM Conf.*, ACM Press, 1977, pp. 153-162.

Peter M. Maurer *is an associate professor of computer science at Baylor University. His research interests include component-level programming, VLSI design, computer architecture, and parallel processing. Maurer received a PhD in computer science from Iowa State University. He is a member of the IEEE Computer Society, the IEEE, and the ACM. Contact him at peter_maurer@baylor.edu.*



PURPOSE The IEEE Computer Society is the world's largest association of computing professionals, and is the leading provider of technical information in the field.

MEMBERSHIP Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE
The IEEE Computer Society's Web site, at **www.computer.org**, offers information and samples from the society's publications and conferences, as well as a broad range of information about technical committees, standards, student activities, and more.

OMBUDSMAN Members experiencing problems—magazine delivery, membership status, or unresolved complaints—may write to the ombudsman at the Publications Office or send an e-mail to help@computer.org.

CHAPTERS Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION
To obtain more information on any of the following, contact the Publications Office:

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

To check membership status or report a change of address, call the IEEE toll-free number, +1 800 678 4333. Direct all other Computer Society-related questions to the Publications Office.

PUBLICATIONS AND ACTIVITIES

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, industry trends, and product reviews.

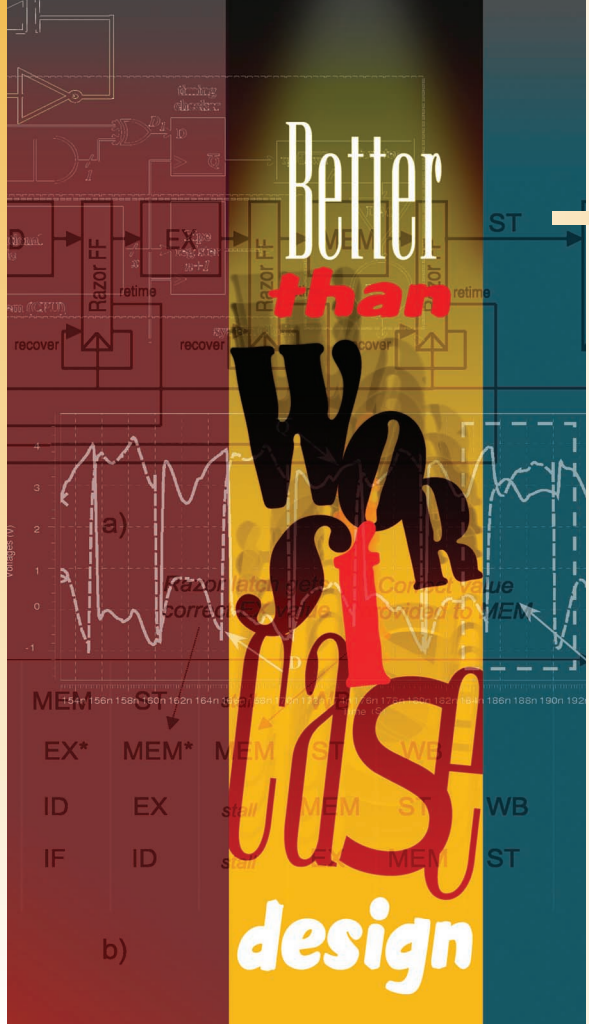
Periodicals. The society publishes 12 magazines and 10 research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 160 titles every year.

Standards Working Groups. More than 200 groups produce IEEE standards used throughout the industrial world.

Technical Committees. Thirty TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.



THEME ISSUE INTRODUCTION

We May Need a New Box

Meeting emerging computer design industry challenges requires finding a balance between continuing to apply old technology beyond the point where it is workable and prematurely deploying new technology without knowing its limitations.

Bob Colwell

We hardware engineers believe in some things that aren't quite true. We call ourselves electrical engineers because we think we understand electrical current—the flow of electrons in conductors, semiconductors, and other materials, under interesting external conditions such as electric or magnetic fields and temperature gradients. The reality is that we have a practical working understanding of what large numbers of electrons will do en masse. But for single electrons, we can't even say where they are, and physicists would chide us for asking the question. Our knowledge is statistical, not absolute. We stack the odds in our favor by employing very large numbers of electrons in our circuits.

As always in engineering, there are limits to what we know. We're used to this—it feels natural. But to physicists, defining the border between what is known and what is unknown is irresistible. They may fret over whether Schrödinger's cat is alive, dead, or both; engineers will look at the airtight box, then at their watches, wait, and then confidently say, "Dead. What's the next item on the agenda?"

EXCEEDING THE SAFE ZONE

Our pragmatism can work against us. Because our collective knowledge is bounded—yet we rightly place great value on our experiences (successes and failures)—we are constantly in danger of accidentally going beyond that safe zone. One way to hurt ourselves is to prematurely deploy new implementation technology before we've comprehended its unique limitations; the other is to continue applying old technology to new problems beyond the point where it is workable.

Universities teach students to practice a conservative approach to engineering called "worst-case design." A bridge designer can tell you the heaviest load a bridge has been designed to carry; a locomotive designer can tell you the gross weight of a railroad engine. Knowing those two things, you can quickly determine whether a train with three engines, crossing a bridge that spans three engine lengths, is likely to reach its intended destination or take a fast vertical detour. Those engines could be lighter than specified, or the bridge designer may actually have designed the bridge to two or three times the actual rated load. You don't necessarily know those safety margins, but you do know that if they tell you a worst-case number, you can reasonably expect to use that number successfully. You should also check the assumption that the engine is the heaviest part of the train and ensure that the bridge isn't covered with heavy ice.

BETTER THAN WORST-CASE DESIGN

Digital design engineers are used to reading data

Published by the IEEE Computer Society

books or cell libraries that tell them the worst-case behavior of the components they are designing into a system. Knowing how fast or slow a given component will be under normal operating conditions, the designer can stack these numbers end to end to get the minimum and maximum propagation delay times through a logic chain.

But is that really possible? Silicon chips are made in a chemical/mechanical fabrication process, tested, and assigned to various "speed bins." These chips don't all run at the same speed, despite the best attempts by designers and production engineers to make them do so. Instead, a distribution of speeds occurs, with many chips running at a middling clock rate, a few much faster, and some much more slowly—or not at all, which is yet another statistical distribution governing the process.

So when we talk about the worst-case design numbers for a given chip, we're really referring to some propagation delay at which the chip's manufacturer hopes to achieve enough yield for a profitable part. Most manufacturers do not test every chip across all axes—thermal temperature, clock rates, loading, and full test vector set—because that is too expensive, and experience suggests it's unnecessary. Instead, they do statistical sampling, using a small number of real chips to predict the behavior of all the chips. Then they add a safety factor to cover what they don't know, but they don't tell you what that safety factor is. They don't know precisely what that margin is; that is the nature of a safety margin, and its existence is a universal constant throughout all engineering.

We computer designers have been living well for the past 35 years. Except for the advanced Schottky debacle of the late 1980s, chips have behaved as their worst-case parameters suggested, and designs incorporating those chips and observing those parameters are likely to work as intended. So we've designed incredibly complicated microprocessors with a hundred million transistors, most of which must work correctly every single time or the processor will make a nonrecoverable error.

There are clouds gathering on the horizon, though.

IN THIS ISSUE

As Naresh R. Shanbhag points out in "Reliable and Efficient System-on-Chip Design," virtually all of the technology development trends today point in the wrong direction: Thermal and leakage power are growing exponentially; system noise sources are increasing, while voltage output is decreasing (and noise margins along with it); yet there is still a strong desire to improve system performance.

Microprocessor-circuit engineers have been grappling with noise for at least a decade, but so far architects and microarchitects have been able to ignore it. While designing in the presence of noise may be novel to us in the computer field, it's a staple item in the communications field, and they may have useful techniques for us to consider. Shanbhag's solution to the problem is to employ information theory to determine achievable bounds on energy/throughput efficiency and to develop noise-tolerance techniques at the circuit and algorithmic levels to approach these bounds.

A safety margin is a universal constant throughout all engineering.

In "Going Beyond Worst-Case Specs with TEAtime," Gus Uht proposes an idea whose time may be here. TEAtime suggests that if critical paths in a design were shadowed by checker circuits, carefully engineered to be slightly slower than the critical path itself and designed to detect failures in the checker circuit, the resulting machine could run substantially faster or at lower supply voltages.

While Uht shows a way to shave operating margins while still maintaining error-free operation, in "Making Typical Silicon Matter with Razor," Todd Austin and colleagues propose that "if you aren't failing some of the time, you aren't trying hard enough," a sentiment that I have seldom seen (purposely) applied to an engineering endeavor! The Razor design incorporates self-checking circuits at the flip-flop level to permit pushing clock frequency and supply voltages beyond normal worst-case levels. Razor's premise is that monitoring a microprocessor's real-time operation and recovering from detected errors would effectively subtract out the accumulated baggage of most of the safety margins implicit in all levels of the machine.

"Speeding Up Processing with Approximation Circuits" by Shih-Lien Lu addresses the general question of how to design circuits and functions to accomplish their tasks without the burden of worst-case design.

Some serious challenges are emerging in the computer design industry. These articles provide a tantalizing and sometimes scary look at a possible shape of things to come. You've heard of thinking outside the box? You can't even see the box from here. ■

Bob Colwell, *Intel's chief IA32 architect through the Pentium II, III, and 4 microprocessors, is now an independent consultant. Contact him at bob.colwell@comcast.net.*

Reliable and Efficient System-on-Chip Design

The recent emergence of noise and increase in process variations raise serious questions about our ability to design reliable and efficient computing systems using nanometer processes. A communication-theoretic design paradigm has been proposed as the solution.

Naresh R. Shanbhag
University of Illinois at Urbana-Champaign

Power dissipation is a concern in both microprocessors and communication systems. High power dissipation increases the substrate temperature of integrated circuits, which increases leakage currents, reduces performance and battery life for mobile applications, and adversely impacts material reliability.

Supply noise (bounce and IR drops), leakage, and interconnect noise (coupling) impact signal phase and amplitude, while process variations result in uncertainty and create a mismatch between signal paths. Both noise and process variations impact reliability, causing logic errors that can result in system failure.

To increase processor performance, the microprocessor industry is driving the scaling of feature sizes into the deep-submicron (DSM) and sub-100-nanometer regime. Unfortunately, power/performance-enhancing design techniques only aggravate the reliability problem. For example, the popular supply voltage scaling technique reduces power, but it does so at the expense of noise immunity.

Although researchers have developed complex power management systems and expensive packaging schemes, the recent emergence of noise and the dramatic increase in process variations have raised serious questions about the capacity to use nanometer process technologies to design reliable and low-power/high-performance computing systems. These concerns put at risk the affordability of microsystems and jeopardize the semiconductor industry's ability to extend Moore's law into the nanometer realm.

The design and electronic design automation (EDA) communities must work closely with the process engineering community to address these problems. Researchers in academia and industry have taken major steps in this direction by establishing the multiple-university Gigascale Silicon Research Center and the Center for Circuit & System Solutions. Both centers, funded through the Microelectronics Advanced Research Corporation (MARCO) by the Semiconductor Industry Association and the Defense Advanced Research Projects Agency, initiated reliability research thrusts beginning in 2003.

Microprocessor designs must achieve high performance and energy efficiency in the presence of noise. A *communication-theoretic* paradigm[1] for reliable and efficient system-on-chip (SoC) design views integrated microsystems as miniature communication networks operating in the presence of noise. First proposed in 1997, this paradigm has evolved into two distinct but related areas of research:

- information-theoretic techniques for determining the lower bounds on energy efficiency in the presence of noise,[1-4] and
- circuit[5] and algorithmic noise-tolerance techniques[6] to approach these bounds.

The 2003 *International Technology Roadmap for Semiconductors* (http://public.itrs.net/Files/2003ITRS/Home2003.htm) echoes the need for a communication-centric SoC design paradigm and identifies error tolerance as a design challenge.

Published by the IEEE Computer Society
0018-9162/04/$20.00 © 2004 IEEE

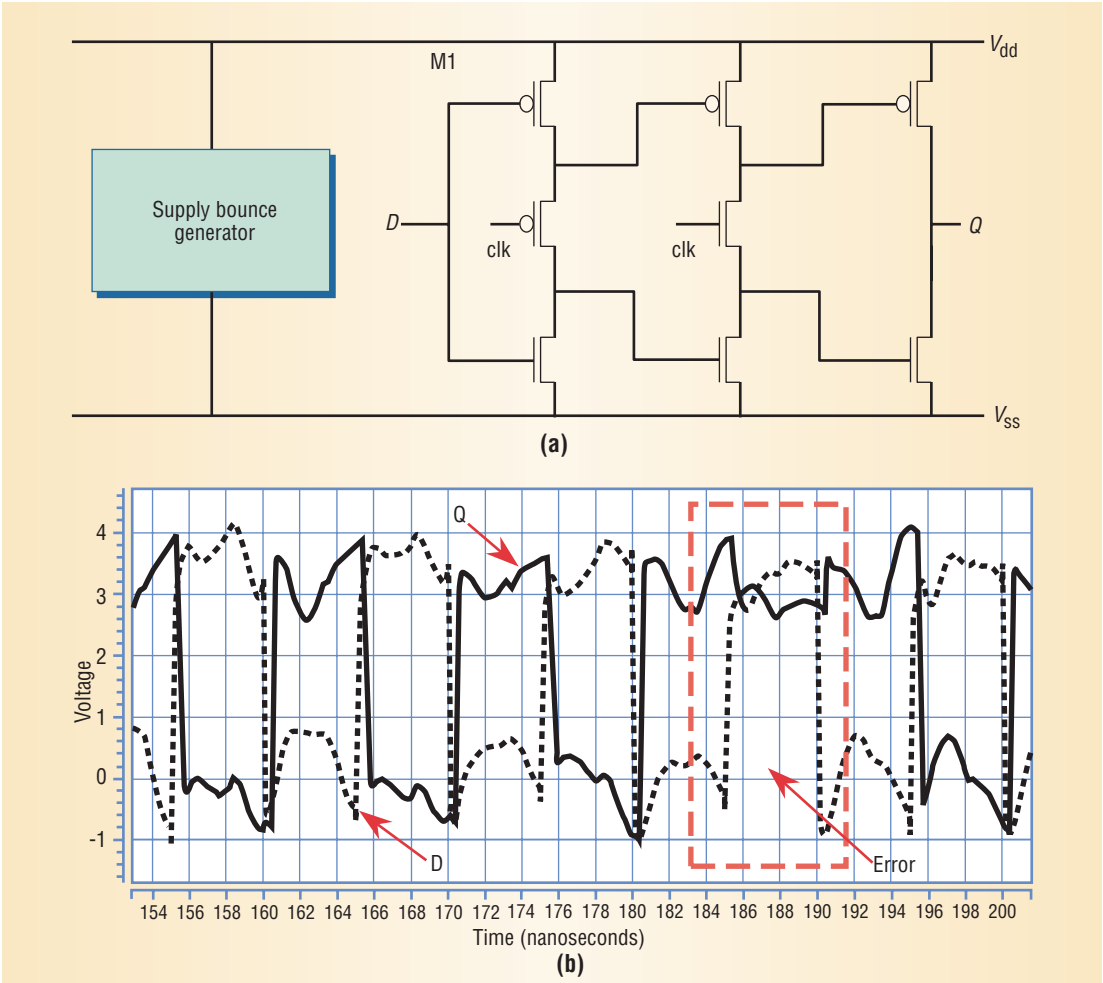


Figure 1. Noise due to supply bounce. (a) D-register circuit schematic. (b) Input and output waveforms. Q and D have the same polarity in the dotted box, indicating a logic error.

DSM NOISE

In DSM circuits, noise is any disturbance that drives node voltages or currents away from a nominal value, causing permanent as well as intermittent errors.[7] If storage elements capture these errors, the result is an observable loss in functionality. Numerous masking mechanisms can prevent an error at the output of a logic gate from propagating further. Increased delay and accidental discharge/charge of dynamic nodes are common mechanisms for such failures.

Figure 1 illustrates an error at the output of an edge-triggered register. When input D is at logic 1 and an inductive kick raises the supply node above $V_{dd} + |V_{tp}|$, where V_{dd} is the supply voltage and V_{tp} is the positive-channel metal-oxide-semiconductor (PMOS) device threshold voltage, the topmost PMOS M1 in the input stage in Figure 1a will activate and cause the logic error indicated in Figure 1b.

The probability of this error event increases with complexity and with the reduction in the threshold voltage. A typical 0.13-µm CMOS process has device threshold voltages in the 200-300 millivolt range, and it is not unusual for the power supply grid to generate supply bounce of a few hundred millivolts. Thus, the types of errors shown in Figure 1 are not unusual.

Accurately modeling the numerous noise sources is difficult. *Noise mitigation* and *noise tolerance* are two distinct ways to handle DSM process noise. The EDA industry favors noise mitigation, which involves developing noise-analysis tools that identify hot spots and then having designers mitigate the impact of noise by focusing on those areas.

Although noise mitigation is an obvious solution to the reliability problem, it is fundamentally inefficient in terms of energy conservation. In contrast, noise tolerance, which is central to the communication-theoretic SoC design paradigm, requires designers to develop circuit and system design techniques that are inherently tolerant to noise and errors. When the design must achieve both energy efficiency and reliability, noise tolerance is the preferred approach.

RELIABLE SOC DESIGN TECHNIQUES

Present-day SoC design techniques are analogous to those used in the design of communications systems more than 50 years ago.

Claude Shannon[8] first proved the feasibility of reliable data transmission over noisy communication links in 1948. Subsequently, communications system designers mastered the science of develop-

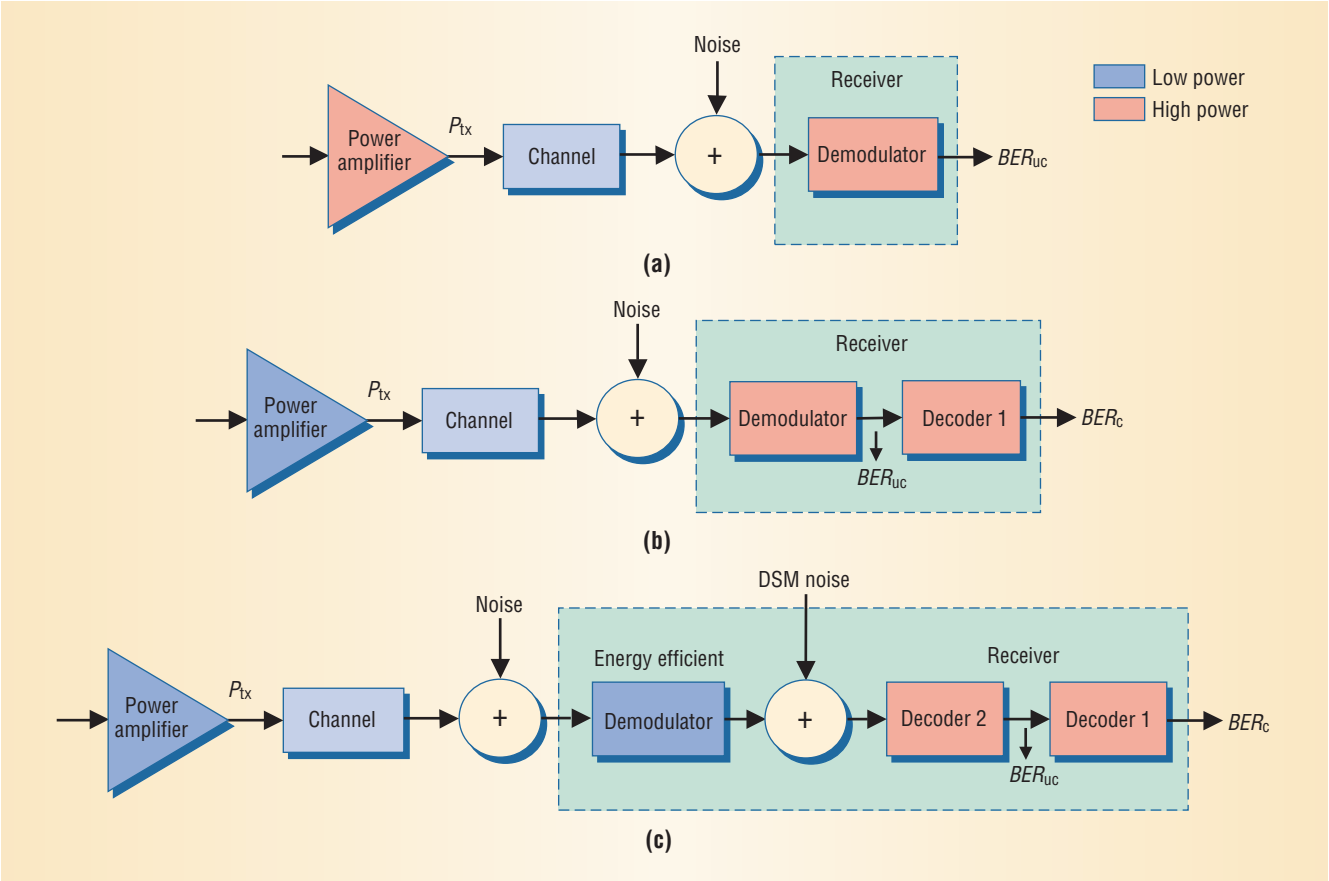


Figure 2. Communication-link design. (a) High transmit power (P_{tx}) will achieve a bit error rate (BER_{uc}) of 10^{-10} but is costly in transmit energy per bit. (b) Error-control coding makes it possible to achieve the required BER_c with low transmit power. (c) Adding error control for both channel and deep-submicron (DSM) noise provides reliability while reducing communication and computational power dissipation.

ing high-speed data transmission techniques that operate under a transmit power constraint and in the presence of noise.

SoC design must likewise be based on an integrated view of system-level reliability and energy efficiency. However, SoCs currently achieve reliability by generating signal power that exceeds noise, an approach that is extremely inefficient in terms of energy consumption.

Figure 2a illustrates a simple communication link, which consists of a power amplifier at the transmitter, the physical channel (copper, cable, optical fiber, or air), and a demodulator at the receiver; the transmitter also includes a modulator, not shown, that feeds into the power amplifier. Channel noise causes errors at the demodulator's output.

Reliable links must achieve a specific end-to-end bit error rate; a BER less than 10^{-10} is typical for data communication links. Pumping up the transmit power so that the received signal power overwhelms the channel noise power could easily achieve the required BER, but at the cost of high energy consumption as measured in terms of transmit energy per bit.

Figure 2b illustrates an alternative based on the design philosophy of correcting rather than avoiding errors. In this approach, a channel encoder at the transmitter, not shown, adds error-protection bits to the information-bearing data bits and then feeds the resulting redundant data stream into the modulator, which feeds into the power amplifier. This reduces the transmit power by many decibels.

The demodulator output is targeted to achieve a BER of roughly 10^{-3} to 10^{-4}, while the decoder output provides the required BER of 10^{-10}. In addition to meeting the BER reliability requirements, this approach is also significantly more energy efficient in terms of transmitted energy per bit. The question is whether designers can use a similar concept to reduce energy consumption in SoC designs, especially for computation.

Dramatically reducing the supply voltage would reduce speed and noise margins in the demodulator, increasing its energy efficiency. As Figure 2c shows, inserting a second decoder would correct the additional errors that this would cause. If the decoder is small relative to the demodulator, the result is a low-power link that significantly reduces both the communication and computational power dissipation, matching the demodulator's reliability with the reliability of the data it recovers.

NOISE TOLERANCE

Researchers[1-4,9] have shown that it is possible to compute reliably in a logic gate implemented in DSM process technology that is subject to random or unpredictable noise. By applying information theory to circuits, it is possible to determine the minimum energy needed to compute a task in the presence of noise.

Energy-Efficiency Bounds for DSM Circuits

Entropy, a key function in the application of information theory to circuits, is defined as

$$h(p) = -p\log_2(p) - (1-p)\log_2(1-p), \quad (1)$$

where $0 \le p \le 1$. The entropy function has the shape of an inverted parabola with minima of $h(0) = 0$ and $h(1) = 0$ and a maximum of $h(0.5) = 1$.

Assume that a two-input AND gate must process data at a rate of f_s bits per second from a data source that generates 1 and 0 with a probability of 1/2, subject to the following parameters: The nominal supply voltage V_{dd} is 1.5 volts, the noise standard deviation is σ_n, the load capacitance C_L is 50 femtofarads (fF), and the MOS devices' transconductance k_m is 200 μA/V^2.

To determine the minimum energy required to implement this gate given the data-rate requirements and noise and process parameters, it is necessary to first abstract out the impact of noise by a single parameter ε, the probability of the AND gate making an error. The relationship between ε and V_{dd} for a noise source that has a Gaussian distribution with a zero mean and standard deviation σ_n is given by

$$\varepsilon = Q\left(\frac{V_{dd}}{2\sigma_n}\right); Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2}\,dy \quad (2)$$

where the noise source is assumed to appear at the gate input as noise voltage V_n. The gate makes an error if V_n exceeds the gate-decision threshold voltage $V_m = 0.5\ V_{dd}$. For example, if V_{dd} is equal to 1.5 volts and σ_n is equal to 500 millivolts, Equation 2 results in 0.067—on average, 67 errors will occur in 1,000 outputs. Note that ε is a monotonically decreasing function of V_{dd}. This trend is consistent with the notion that circuits' noise immunity increases with the supply voltage.

Employing the approximate relationship

$$f_c = \frac{k_m V_{dd}}{C_L}$$

and information-theoretic concepts, the following expressions can be used to determine dynamic power dissipation and the lower bound on supply voltage for reliable operation:

$$P_d = tC_L V_{dd}^2 f_c = h^{-1}\left[\frac{RC_L}{k_m V_{dd}} + h(\varepsilon)\right] V_{dd}^3 k_m; \quad (3)$$

$$V_{dd-min} = \frac{h(p_y) f_s C_L}{k_m\left[1 - h(\varepsilon)\right]}; \quad (4)$$

where $h^{-1}[\ ]$ is the inverse of the entropy function and $R = h(p_y)f_s$—p_y being the probability of observing a 1 at the output—is the information transfer rate. Equation 3 unveils a tradeoff between the transition activity t and the supply voltage: As the supply voltage decreases, the inverse entropy term representing t increases, thereby offsetting reduction due to the cubic term.

Figure A plots Equation 3 as a function of V_{dd} when σ_n is equal to 300 millivolts and R is 1 Gbit per second. Note that the supply voltage at which dynamic power dissipation is minimized (V_{dd-opt}) is greater than the minimum supply voltage for which a reliable implementation of the AND gate exists (V_{dd-min}). Reliability and energy efficiency are thus elegantly linked together. In Figure 3, V_{dd-opt} is equal to 0.8655 volts and V_{dd-min} is 0.6303 volts; the energy consumed when V_{dd} is equal to V_{dd-opt}—the lower bound on energy dissipation—is E_{b-min}, or 15.5 femtojoules per bit.

As equations 2 and 3 show, noise ε and the information transfer rate R play a key role in determining the lower bounds on power dissipation and the supply voltage. Past attempts at quantifying such lower bounds have ignored this dependence and thus are incomplete.

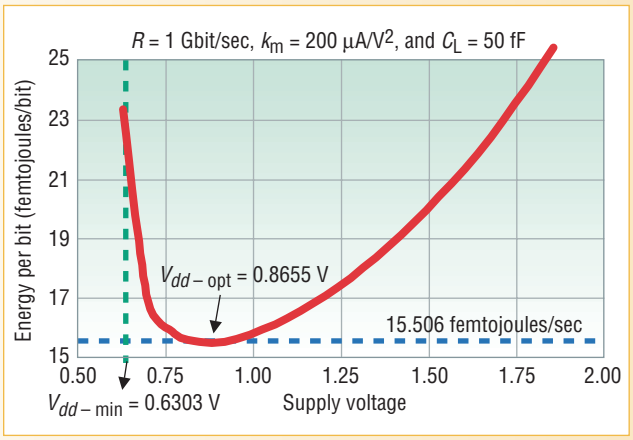


Figure A. Lower bounds on computational energy per bit for single output logic gate. The supply voltage at which dynamic power dissipation is minimized (V_{dd-opt}) is greater than the minimum supply voltage for which a reliable implementation of the AND gate exists (V_{dd-min}).

A noisy circuit with *information transfer capacity* C bits per second can reliably process a data source with an *information transfer rate* of R bits per second provided C is greater than R. The lower bound on energy efficiency is obtained when the capacity C is quantifiably close to R. The "Energy-Efficiency Bounds for DSM Circuits" sidebar describes this relationship in more detail. In the specific context of bus transmission, achievable energy efficiencies are a factor of 24 times below current-day systems.

Research has also shown that it is possible to compute reliably even when the signal and noise powers are comparable.[1-4,9] This implies that if computation is to occur near the limits of energy efficiency, noise tolerance is the correct design philosophy to optimize energy consumption while maintaining reliability.

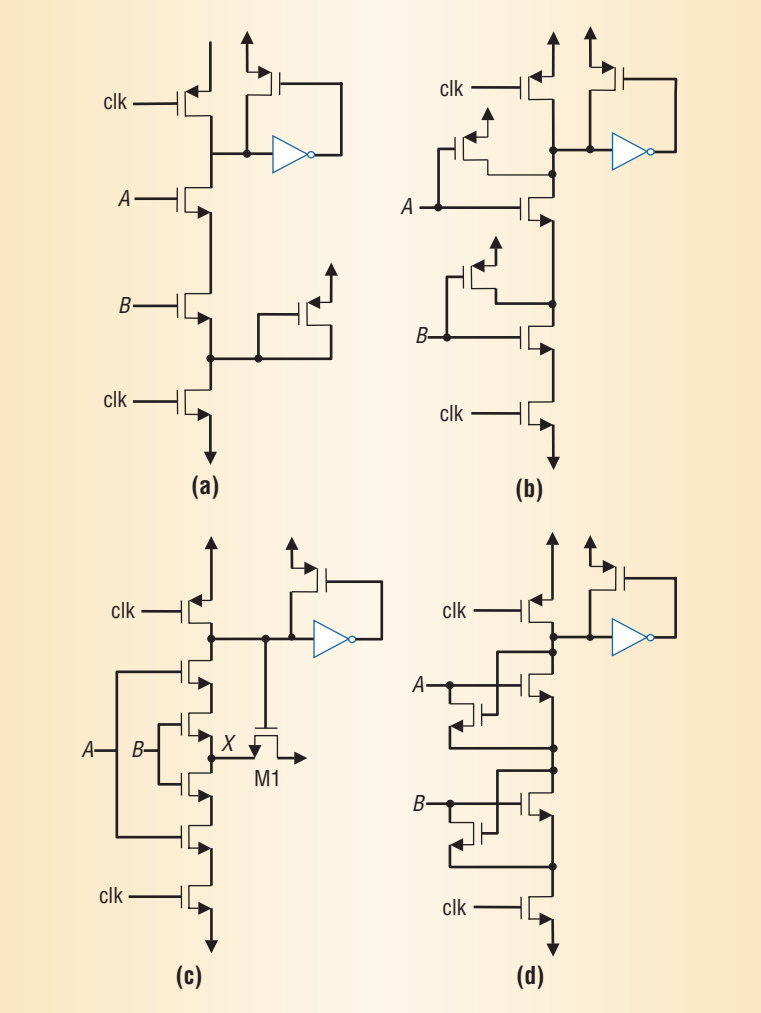


Figure 3. Noise-tolerant circuit techniques: (a) PMOS pull-up, (b) CMOS inverter, (c) mirror, and (d) twin-transistor.

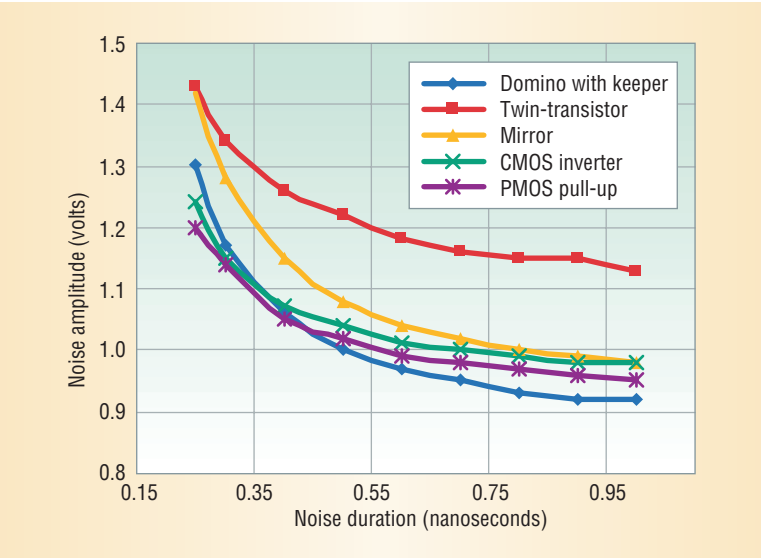


Figure 4. Noise-immunity curves for a two-input AND gate implemented using five dynamic circuit styles in a 0.18-μm, 1.8-V CMOS process. All of the circuit styles are designed to operate at a delay of 280 picoseconds when driving a load of 20 femtofarads.

Noise-tolerant circuit design

Dynamic circuits provide a convenient platform for studying the tradeoffs between energy efficiency and noise tolerance. Enhancing noise tolerance in a circuit carries an energy and power penalty or tax; effective techniques minimize this tax when they achieve a specified level of noise tolerance.

Figure 3 illustrates four dynamic circuit styles designed to provide noise tolerance: positive-channel MOS (PMOS) pull-up, complementary MOS (CMOS) inverter, mirror, and twin-transistor.

The *PMOS pull-up* technique[10] utilizes a pull-up device to increase the source potential of the negative-channel metal oxide semiconductor (NMOS) device, thereby increasing the transistor threshold voltage V_{tn} and hence the switching threshold voltage V_{sw} of the gate during the evaluation phase. This technique suffers from large static power dissipation.

The *CMOS inverter* technique[11] utilizes a PMOS transistor for each input, thereby adjusting V_{sw} to equal that of a static circuit. This technique cannot be used for NOR-type circuits as certain input combinations can generate a direct path from supply to ground.

The *mirror* technique utilizes two identical NMOS evaluation networks and one additional NMOS transistor M1 to pull up the source node of the upper NMOS network to $V_{dd} - V_{tn}$ during the precharge phase, thereby increasing V_{sw}. This technique guarantees zero DC power dissipation, but a speed penalty is incurred if the transistors are not resized.

The *twin-transistor* technique[5] represents the state of the art in noise-tolerant dynamic circuit design. It employs an extra transistor for every transistor in the pull-down network to pull up the source potential in a data-dependent manner. The twin-transistor technique consumes no DC power and has a limited impact on speed and power.

Noise immunity curves

Conservative static noise margin metrics do not account for the inherently low-pass nature of logic circuits, which can filter out noise pulses that either have a small amplitude or short duration. Thus, a comparison of circuit styles requires using metrics such as the noise immunity curves (NICs) shown in Figure 4.

A digital gate's NIC is a locus of tuples (V_n, T_n), where V_n is the noise amplitude in volts and T_n is the noise duration in seconds, representing noise pulses that generate a logic error for that gate. Noise pulses above the curve are guaranteed to gen-

erate logic errors, while those below are guaranteed not to do so.

A logic error is said to occur when the output crosses a predefined voltage threshold, usually halfway between the supply rails, or when the output glitch amplitude equals the input noise-pulse amplitude. The *average noise threshold energy*[5] is a convenient metric that can be derived from the NIC by averaging the energy of the noise pulses that cause an error. Normalizing the ANTE with the energy consumption provides the NANTE[5] metric, a measure of the noise-tolerance circuit technique's effectiveness.

Table 1 quantifies the ANTE, energy, and NANTE metrics for each of the four noise-tolerant techniques as well as a conventional domino design when implemented in a 0.18-µm, 1.8-volt CMOS process. The twin-transistor technique has the best ANTE and NANTE metrics, indicating that it provides the highest noise immunity per unit of energy consumption. Both the mirror and twin-transistor techniques have been proved experimentally in the past via the design and test of prototype chips in 0.35-µm CMOS technology.

More research is needed to develop dynamic circuit styles that are not only tolerant to various noise sources but also have low-noise-generation features. Circuit-level techniques are not sufficient, especially when energy efficiency also is a concern. Noise tolerance techniques are required at the architectural, algorithmic, and system levels of the design hierarchy.

Table 1. Noise tolerance versus energy efficiency.

Dynamic circuit technique	Average noise threshold energy (picojoules)	Energy (picojoules)	Normalized ANTE
Conventional domino	586.6	0.2688	2,182
Twin-transistor	859.1	0.2862	3,002
Mirror	633.4	0.3158	2,006
CMOS inverter	622.3	0.2752	2,261
PMOS pull-up	606.2	0.4826	1,256

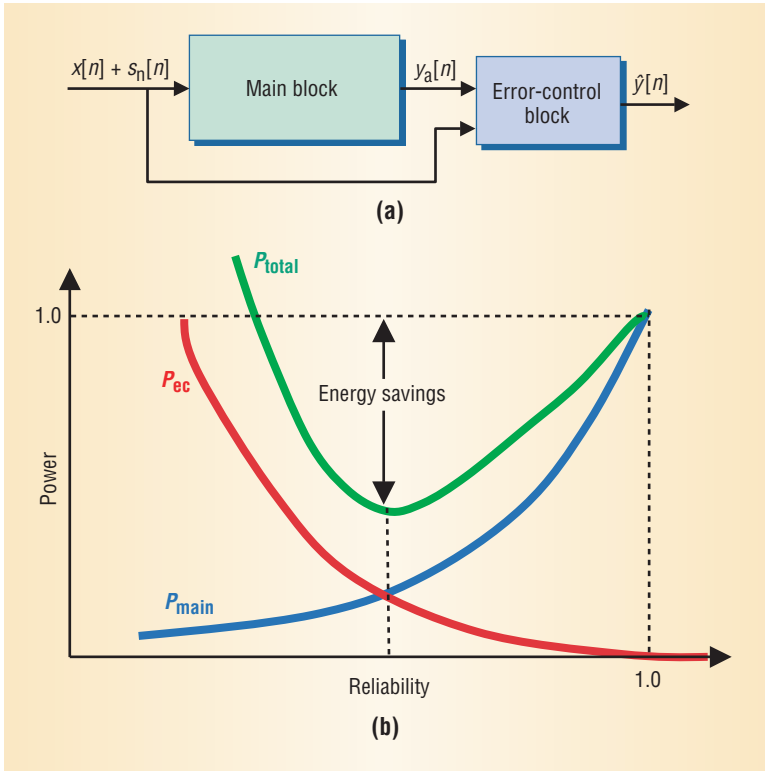


Figure 5. Algorithmic noise tolerance. (a) An ultra-energy-efficient main block executes most required computations; a reliable and thus energy-inefficient error-control block detects and corrects intermittent errors by the main block. (b) As the main block's energy efficiency increases, its reliability decreases, necessitating the use of increasingly complex error control.

ALGORITHMIC NOISE TOLERANCE

The key idea behind *algorithmic noise tolerance* (ANT),[6] illustrated in Figure 5a, is that an ultra-energy-efficient main block executes most of the required computations. The main block can make intermittent errors as long as they occur infrequently.

An *error-control block*, which is reliable and thus energy inefficient, detects and corrects these errors. As Figure 5b shows, as the main block's energy efficiency increases, its reliability decreases, necessitating the use of increasingly complex error control. Thus, the error-control block's power dissipation increases.

The total power dissipation of the main block and the error-control block reaches its minimum when the main block achieves a specific level of reliability, which is determined by the error frequency and the error control's effectiveness.

ANT techniques

Effective error-control techniques provide a high level of error detection and correction with low hardware complexity. Thus, error-control blocks can have a relaxed delay and power constraint.

Significantly relaxing delay and power constraints permits enhancement of noise immunity in the error-control block. For example, researchers can use the noise-tolerant twin-transistor technique[5] along with noise-analysis and noise-mitigation techniques to design a robust error-control block that has minimal impact on speed and power. Instead of focusing on the entire SoC, a designer need only ensure that the few error-control blocks in a complex design are robust at the circuit level.

Developers of error-control techniques for signal-processing kernels can use statistical performance metrics, such as the signal-to-noise ratio, to exploit the signals' statistical structure. In such

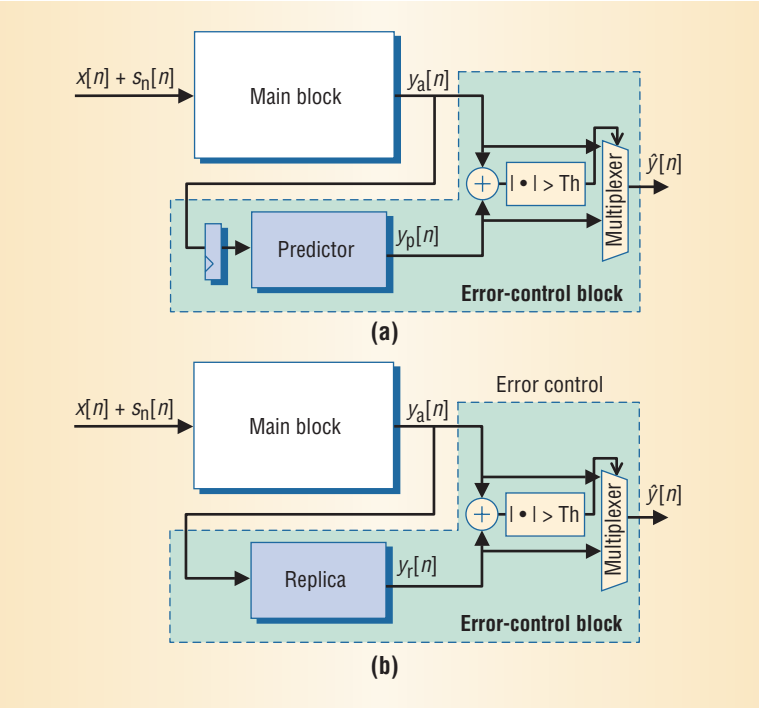


Figure 6. Algorithmic noise-tolerance techniques. (a) A predictor uses the past outputs to generate a statistical estimate of the main block. (b) Reduced-precision redundancy employs a replica of the main filter as an estimator.

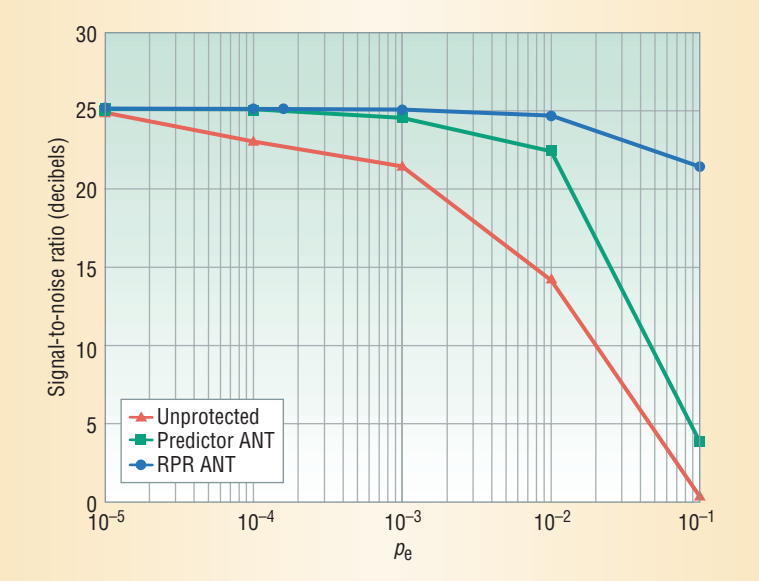


Figure 7. Predictor and reduced-precision redundancy ANT technique performance in the presence of random noise. The signal-to-noise ratio improves by 10 decibels even when each output bit of the filter is independently flipped at an average rate of once every 1,000 samples.

cases, designers can use signal-estimation techniques for error control.

The following examples all assume that the main block is a digital filter that makes intermittent errors.

In the *predictor* technique, shown in Figure 6a, another filter (predictor) uses the past outputs $y_a[n-1], y_a[n-2], \ldots, y_a[n-N_p]$ to generate a statistical estimate $y_p[n]$ of the main block. Designers can use standard statistical signal-processing techniques to determine optimal predictor coefficients that minimize the mean-squared error between the predictor and main filter outputs. The error-control block detects errors by comparing the predictor and main filter outputs. When the difference between the two outputs exceeds a prespecified threshold, the error-control block declares an error. In the event of an error, the error-control block selects the predictor output $y_p[n]$. Both error detection and correction are approximate, which is reasonable given the interest in maintaining the signal-to-noise ratio.

Figure 6b illustrates *reduced-precision redundancy*, another simple, yet effective, error-control technique. RPR employs a replica of the main filter as an estimator. The error-detection and -correction steps are the same as those in the predictor technique except that the prespecified threshold must be greater than the quantization noise floor.

Figure 7 shows predictor and RPR performance when the main filter output bits are randomly flipped with frequency p_e. The signal-to-noise ratio improves by 10 decibels even when each output bit of the filter is independently flipped at an average rate of once every 1,000 samples.

Voltage overscaling

One way to study the tradeoff between energy efficiency and reliability is to reduce the supply voltage below the minimum required for correct operation. This *voltage overscaling*[6] results in delay violations, causing output errors whenever the user applies an appropriate input sequence. VOS, which has recently been used in the design of reliable low-power microprocessors,[12] improves energy efficiency beyond what present-day supply scaling can achieve.

Although VOS errors are systematic, they are modeled as being random when using ANT techniques, which are known to be effective for random errors. In addition, error-control techniques that exploit the systematic nature of VOS errors are too complex to be of any practical use.

A voltage-overscaled digital filter chip incorporating the predictor technique and implemented in a 0.35-µm CMOS process has demonstrated up to 70 percent savings in energy over a filter operating at critical supply voltage.

ANT-based techniques can be used instead of triple-modular redundancy (TMR) to provide robustness against soft errors due to particle hits. Much greater energy efficiencies are achievable by employing one main filter block and two estimators. In contrast, TMR would require three main

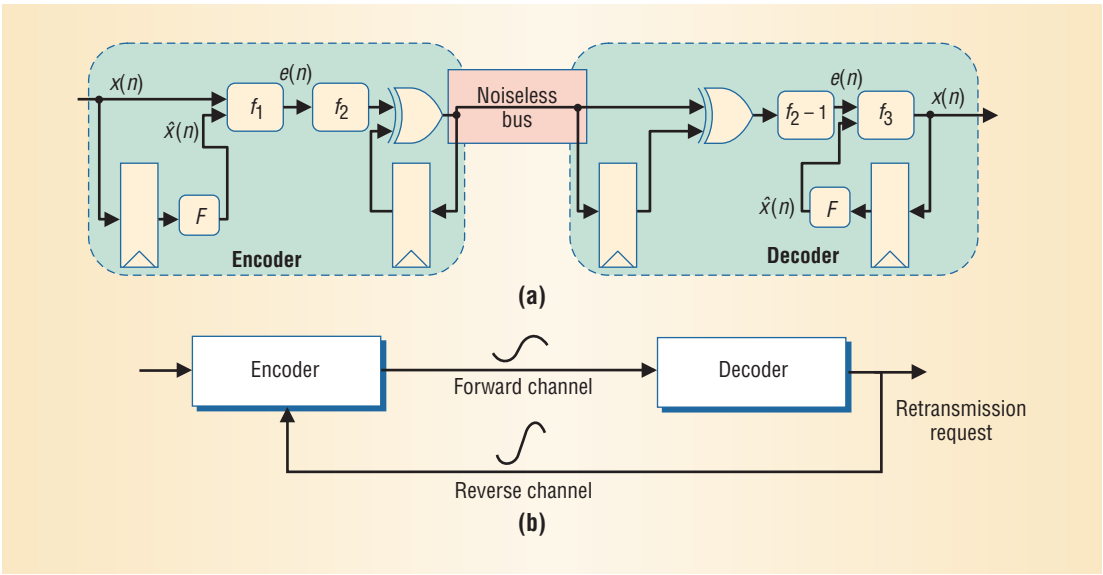


Figure 8. Bus coding. (a) A bus-coding framework based on source coding consists of a predictor F, differentiator f_1, and mapper f_2. (b). Use of Hamming and Reed-Muller codes along with a reverse retransmission request channel reduce signaling levels in the frequently used forward channel.

filter blocks, resulting in very high overhead. In such a case, the assumption of an error-free error-control block can be relaxed.

NOISE-TOLERANT BUS TRANSMISSION

Buses are key SoC components. Coupling between adjacent wires, supply bounce at the receiver, and other sources can cause noise in buses. Energy consumption in buses occurs mainly due to transitions on bus lines, including charging and discharging the self and coupling capacitances.

Early bus power-reduction techniques ignored coupling, focusing instead on reducing transition activity in individual bit lines. As Figure 8a shows, a bus-coding framework[13] based on source coding—for example, video compression employed in multimedia communication networks—has three key elements. The predictor F can be an identity or an increment function; the differentiator f_1 can be an XOR or a subtractor; and the mapper f_2 can employ a probability-based mapping, value-based mapping, inversion, or identity function.

Assigning different functionalities to F, f_1, and f_2 results in a family of coding schemes demonstrating this framework's power. For example, the following assignment can derive the well-known *bus-invert* scheme from this framework: F = identity, f_1 = XOR, and f_2 = inversion. Another useful coding scheme for address buses is *INC-XOR*, obtained by the following assignment: F = increment, f_1 = XOR, and f_2 = identity.

These and other similar techniques that focus on reducing transition activity in individual bus lines ignore the problem of coupling found in DSM processes. Recent work uses coding to minimize transitions on adjacent bus lines, thereby reducing delay. However, none of these techniques address the noise problem.

As Figure 8b shows, the first work addressing noise and energy efficiency in high-speed SoC bus transmission[2] used Hamming and Reed-Muller codes along with a reverse retransmission request channel to reduce signaling levels in the frequently used forward channel. Simulations demonstrated a three- to fourfold power savings, but again this work does not address coupling. Use of error-detection and retransmission was recently proposed for reliable communications in networks- on-a-chip.[14]

A remaining challenge is to develop noise-tolerant bus transmission codes that jointly address coupling, self-capacitance, and noise.

The semiconductor industry faces numerous challenges in developing reliable, energy-efficient SoC designs that are on a par with modern communications systems. Researchers must explore the tradeoffs between reliability and energy efficiency at the device, circuit, architectural, algorithmic, and system levels to develop a reliability-energy "knob" that can be synergistically tuned to meet these requirements at each level of the design hierarchy. Elegant and practical solutions will require the application of coding and communication-theoretic techniques to the design of SoC components. In addition, researchers must develop statistical approaches to design and verification as well as statistical performance metrics. ■

Acknowledgments

This work was supported by the MARCO-sponsored Gigascale Silicon Research Center and National Science Foundation grants CCR-000987 and CCR-0095929. I also thank Krisnamurthy Soumyanath, Ram Krishnamurthy, Tanay Karnik, and Shekhar Borkar at Intel as well as current and former graduate students Sumant Ramprasad, Raj Hegde, Lei Wang, Byonghyo Shim, and Ganesh Balamurugan for their contributions.

References

1. N.R. Shanbhag, "A Mathematical Basis for Power-Reduction in Digital VLSI Systems," *IEEE Trans. Circuits and Systems,* Part II, vol. 44, no. 11, 1997, pp. 935-951.
2. R. Hegde and N.R. Shanbhag, "Towards Achieving Energy Efficiency in Presence of Deep Submicron Noise," *IEEE Trans. VLSI Systems,* vol. 8, no. 4, 2000, pp. 379-391.
3. P.P. Sotiriadis, V. Tarokh, and A. Chandrakasan, "Energy Reduction in VLSI Computation Modules: An Information-Theoretic Approach," *IEEE Trans. Information Theory,* vol. 49, no. 4, 2003, pp. 790-808.
4. P. Abshire and A.G. Andreou, "Capacity and Energy Cost of Information in Biological and Silicon Photoreceptors," *Proc. IEEE,* vol. 89, no. 7, 2001, pp. 1052-1064.
5. G. Balamurugan and N.R. Shanbhag, "The Twin-Transistor Noise-Tolerant Dynamic Circuit Technique," *IEEE J. Solid-State Circuits,* vol. 36, no. 2, 2001, pp. 273-280.
6. R. Hegde and N.R. Shanbhag, "Soft Digital Signal Processing," *IEEE Trans. VLSI Systems,* vol. 9, no. 6, 2001, pp. 813-823.
7. K.L. Shepard and V. Narayanan, "Conquering Noise in Deep-Submicron Digital ICs," *IEEE Design and Test of Computers,* vol. 15, no. 1, 1998, pp. 51-62.
8. C.E. Shannon, "A Mathematical Theory of Communications," *Bell System Technical J.,* vol. 27, nos. 3 and 4, 1948, pp. 379-423, 623-656.
9. S. Winograd and J.D. Cowan, *Reliable Computation in the Presence of Noise,* MIT Press, 1963.
10. G.P. D'Souza, *Dynamic Logic Circuit with Reduced Charge Leakage*, US patent 5483181, Patent and Trademark Office, 1996.
11. J.J. Covino, *Dynamic CMOS Circuits with Noise-Immunity*, US patent 5650733, Patent and Trademark Office, 1997.
12. T.M. Austin, "DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design," *Proc. 32nd Ann. Int'l Symp. Microarchitecture,* IEEE CS Press, 1999, pp. 196-207.
13. S. Ramprasad, N.R. Shanbhag, and I.N. Hajj, "A Coding Framework for Low-Power Address and Data Busses," *IEEE Trans. VLSI Systems,* vol. 7, no. 2, 1999, pp. 212-221.
14. D. Bertozzi, L. Benini, and G. De Micheli, "Low Power Error Resilient Encoding for On-Chip Data Busses," *Proc. 2002 Design, Automation and Test in Europe Conf. and Exposition,* IEEE CS Press, pp. 102-109.

Naresh R. Shanbhag *is an associate professor in the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign. His research focuses on the design and implementation of digital integrated circuits and systems for signal processing and communications. Shanbhag received a PhD in electrical engineering from the University of Minnesota. He is a member of the IEEE Solid-State Circuits Society, IEEE Circuits and Systems Society, and IEEE Signal Processing Society. Contact him at shanbhag@uiuc.edu.*

Going Beyond Worst-Case Specs with TEAtime

The timing-error-avoidance method continuously modulates a computer-system clock's operating frequency to avoid timing errors even when presented with worst-case scenarios.

Augustus K. Uht
University of Rhode Island

Virtually all engineers use worst-case component specifications for new system designs, thereby ensuring that the resulting product will operate under the worst conditions they can envision. However, given that most systems operate under *typical* operating conditions that rarely approach the demands of worst-case conditions, building such robust systems incurs a significant performance cost. Further, classic worst-case designs do not adapt to previous manufacturing conditions or current environmental conditions, such as increased temperature.

The *timing-error-avoidance* prototype provides a circuit and system solution to these problems for synchronous digital systems. TEAtime has demonstrated much better performance than classically designed systems and also adapts well to varying temperature and supply-voltage conditions. TEAtime works by increasing the operating frequency of the system clock until just before a timing error would occur, then slightly decreasing the clock frequency. These changes in clock frequency happen continuously. Low cost, TEAtime involves no software actions.

DIGITAL SYSTEM DESIGN

Most current digital systems are synchronous in that their state changes only in response to transitions of a systemwide clock signal, typically changing on the clock's rising edges, that is, from a logical 0 to a logical 1. Such systems work correctly only when the delay from a clocked part's output—the flip-flop—to the input of the same or other flip-flops is less than the *clock period*, the time between the clock's adjacent rising edges. All synchronous systems designed today use delays that assume worst-case environmental, operating, and manufacturing conditions.

Student engineers are often surprised when their first synchronous systems fail to operate as expected. This usually happens because they used typical part specifications during the system design. Thus, on a hot summer day, for example, before the due date at the end of the semester, the gates will have a greater than typical delay, the system clock period originally specified will be too short, and the system will fail.

The solution, as all practicing engineers know, requires using worst-case part specifications for system design. For example, the delay used for a gate will be its delay at the highest specified temperature and lowest supply voltage. Further, engineers build in even more latitude to allow for manufacturing variations. The resulting worst-case design provides a system that will operate under a wide range of operating and manufacturing conditions.

Unfortunately, given that the system usually operates under typical conditions, the worst-case approach incurs a severe performance cost: The system usually runs at a clock frequency much lower than necessary.

PC game enthusiasts frequently compensate for this decrease in performance by increasing the operating frequency of their systems far above what the manufacturers specify. This practice is dangerous, however, because the only way overclockers can know they have pushed their CPUs too far is when

0018-9162/04/$20.00 © 2004 IEEE

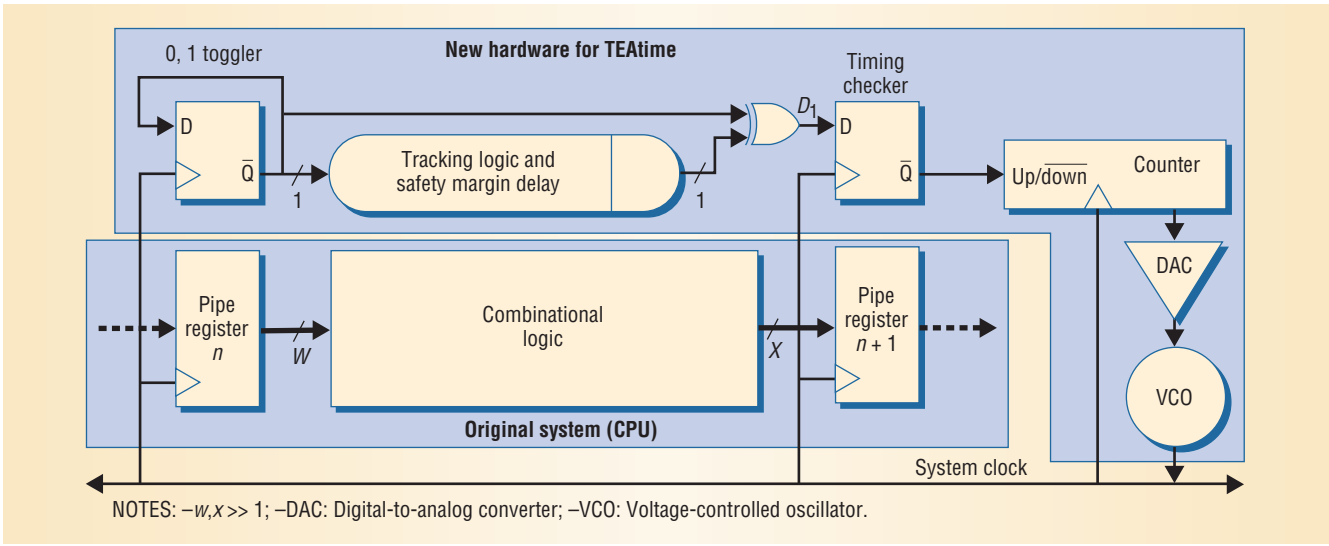


Figure 1. TEAtime CPU or digital system modifications. TEAtime adds a toggler, tracking logic and safety margin delay, timing checker, up-down counter, digital-to-analog converter, and voltage-controlled oscillator.

their systems experience a potentially catastrophic failure. For the casual user, constantly tuning the clock frequency while destroying data and even hardware offers an unsatisfactory tradeoff to enhancing performance beyond stock specifications. The situation becomes much worse when applications more critical to society than the latest shoot-em-up computer game are involved.

ALTERNATIVE APPROACHES

Researchers have sought ways to dynamically increase the clock frequency either without causing timing errors or, if errors do occur, by providing built-in methods for recovering from them. If the resulting solution is always active and always seeks the highest clock frequency possible, the system will also adapt to varying environmental and manufacturing conditions.

In one alternative to TEAtime, a system uses a microcontroller to periodically check the operation of the controller's adder[1] because this component causes the greatest delay in the system. The microcontroller performs the checking by propagating a signal through the adder's entire carry chain by, for example, adding a 1 to all 1s. If this operation provides an incorrect result, the system decreases the clock frequency. Unfortunately, the worst-case path through a system may not be through the adder and thus not be as easy to check. Another disadvantage of this approach is the need to modify the system software for the scheme to work.

A previous technique checked systems by letting errors occur.[2] The system then backed up the state to a known good state, decreased the clock frequency, and continued. This technique more than doubled the hardware needed in the system and was much harder to add to existing designs than anticipated. The recent Razor system[3] also allows errors to occur, then recovers from them. While the additional quantity of hardware needed is small, it is not simple. Also, extra pipeline bubbles and flushes occur, unlike in TEAtime. Large power savings are predicted; improved performance is not a goal.

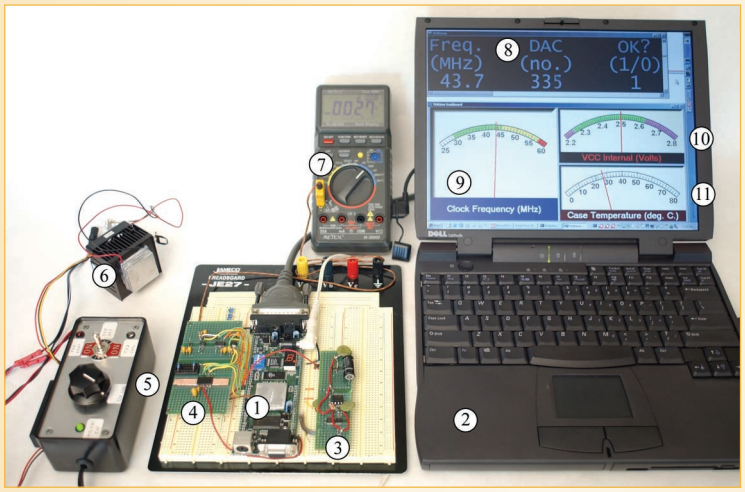
Some techniques modify the clock frequency based on varying constraints,[4] but these systems are *open-loop*: They either do not have a feedback system or do not include the operating frequency or clock period as part of the regulating system. Some systems throttle the clock based on power or temperature,[5,6] but the clock frequency does not increase above the usual worst-case design value.

Self-timed[7] and some other systems[8] use a ring oscillator to mimic the worst-case path delay through the system. Although this method resembles TEAtime, it does not use true tracking logic to mimic the actual circuitry along the worst-case path, thus it does not truly adapt to existing system conditions. This can either produce system failure in the worst scenario or lead to an overly conservative design that does not achieve the high performance that might be possible otherwise.

Asynchronous systems[9] sidestep the problem completely because they do not use a clock. Such systems operate as fast as the gate delays allow. Although this might seem to be an ideal approach, designing robust and error-free asynchronous systems is difficult, and few if any current computer-aided-design tools can generate them.

TEATIME

Figure 1 shows how TEAtime uses tracking logic to mimic the worst-case delay in a synchronous system. Normally, the tracking logic is a one-bit-wide replica of the worst-case path in the system, with a slight delay added to it that provides a safety margin for the system. The flip-flop at the input to the tracking logic is wired as a toggle flip-flop and clocked by the system clock, changing from 0 to 1 and 1 to 0 on alternate cycles. This provides a test signal for the tracking logic during every cycle of operation and ensures that the signal tests both types of transitions. The latter is necessary since



LEGEND

1. FPGA and XESS XSA-100 prototype card
2. PC host
3. Custom oscillator card with DAC and VCO
4. Custom card containing the PC host-controlled FPGA internal supply voltage
5. Custom thermoelectric device controller
6. Thermoelectric device assembly
7. FPGA case temperature meter
8. Controller program command-line window and data display
9. System clock frequency meter
10. VCCint meter
11. FPGA case temperature meter

Figure 2. TEAtime prototype with experimental and demonstration setup. The test computer's system clock has stabilized at about 44 MHz. The normal worst-case specified frequency is about 30 MHz. The system clock frequency meter (9) consists of four regions: White is part of the classical non-TEAtime operating region, green shows better-than-worst-case operating frequency, yellow shows the safety margin, and red shows system failure.

delays for the two transitions can differ.

The tracking logic output then goes through the safety margin delay. Next, the exclusive-OR gate normalizes the test signal for the timing checker flip-flip at the end of the chain—the final version of the test signal, D1, will always change from a 1 to a 0 at the end of the cycle. The timing checker flip-flop also operates with the system clock.

If the timing checker flip-flop latches a 1, this signifies that the system clock period is close to being less than the worst-case path delay, and the system decreases the clock frequency. Conversely, if the flip-flop latches a 0, the clock period is greater than the delay through the worst-case path of the real logic, and the system increases the clock frequency, improving performance.

The timing checker flip-flop output therefore provides the command signal for the system clock generator to increase or decrease the clock frequency. This signal controls the counting direction of the up-down counter. The digital-to-analog converter (DAC) converts the counter output to an analog voltage signal. This signal sets the clock frequency by controlling the voltage-controlled oscillator, and the VCO output becomes the system clock, completing the feedback loop.

Thus, the clock period will never be less than the delay through the tracking logic plus the safety margin delay. Since the system's real logic is as slow or slower than the tracking logic, no timing error will occur in the real logic, which ensures correct system operation.

THE PROTOTYPE

Figure 2 shows the Xilinx field-programmable gate array-based prototype of the TEAtime system. The FPGA contains the TEAtime logic and a test computer. This computer contains a simple 32-bit, five-stage pipelined CPU with forwarding, and the equivalent of a small single-cycle-access cache memory.

The test computer executed a small program continuously during the experiments. This included all typical program constructs such as assignment statements, forward- and backward-conditional branches, and subroutine calls and returns. The program also exercised the pipeline's forwarding paths. The test processor stores the program's results in the cache memory. After every program execution, the PC host controller checks the results for correctness, then resets the results to bogus values before the next execution.

The test computer system clock's nominal worst-case specified operating frequency is about 30 MHz. The unit could be expected to operate at this frequency under even worst-case conditions. This baseline clock frequency was determined by using CAD tools performing worst-case-condition simulations of the test computer executing the test program.

Basic operation and stabilization

The first experiment established TEAtime's basic operational soundness and stabilization properties. With the FPGA's supply voltage held constant, and the test computer operating at room temperature, power was applied to the system. The system clock frequency rose from 25 MHz, the VCO's lowest frequency, then stabilized at about 45 MHz, as Figure 3 shows.

The horizontal line at 30 MHz indicates the approximate baseline worst-case operating frequency. The top subplot shows the value of the control line to the counter driving the DAC. The control is set for constant increases until stabilization, when the control value oscillates between increasing and decreasing frequency. The clock period varies only slightly above and below the delay through the tracking and safety margin logic. From a throughput perspective, TEAtime increases performance by about 50 percent compared with the baseline system under typical operating conditions—exactly the desired results.

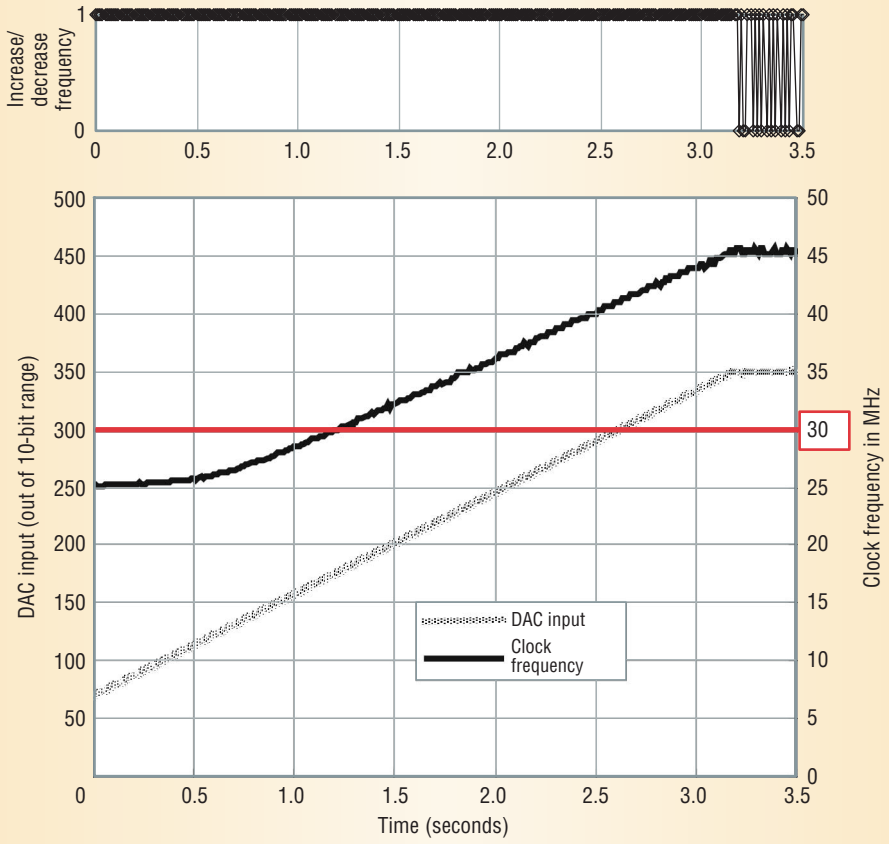


Figure 3. TEAtime's basic operation and stabilization. The clock frequency automatically increases to a high level, then stabilizes. The horizontal line at 30 MHz indicates the approximate baseline worst-case operating frequency. The control is set for constant increases until stabilization, when the value oscillates between increasing and decreasing frequency. The clock period varies only slightly above and below the delay through the tracking and safety margin logic.

TEAtime's adaptability

To examine TEAtime's adaptability to both FPGA temperature and supply-voltage (VCCint) changes, a thermocouple embedded in the center of the aluminum block, which is in turn thermally bonded to the top of the FPGA, measured the FPGA's case temperature—and hence the test computer's.

The FPGA has two supply voltages: one at 3.3 V for its I/O circuitry and one for the FPGA's internal logic, and hence for the test computer's logic as well. For correct operation, Xilinx specifies 2.5 V for VCCint, with an allowed variance from plus 5 percent to minus 10 percent. The green-colored region on the PC host's VCCint meter indicates this range. Only VCCint varied, with the test computer having no direct I/O connections to or from the FPGA chip. The FPGA never operated at above 3 V, which would have been a physically damaging VCCint value.

Figure 4 shows the detailed data for the combined temperature and supply-voltage variations. The supply voltage varied from 2.2 to 2.8 V, while the test computer's case temperature remained constant for each set of voltage data. Even though the operating frequencies varied widely—from 38 MHz to 49 MHz—TEAtime adapted to the existing operating and environmental conditions and always maximized the system clock frequency, within the safety margin delay. The test program executed correctly in all cases. In a perhaps extreme example of this adaptation to environmental conditions, a plastic-wrapped ice cube was placed directly on the FPGA's aluminum block, reducing the case temperature to 3° C. The system still adapted to the existing environmental conditions, functioning correctly and at a high clock frequency.

Tuning the system

The prior experiments used a large safety margin delay, but in this experiment, decreasing the delay and running the system to stabilization with typical conditions dramatically improved the operating frequency to 53 MHz. Time-related performance increased to 43 percent over the baseline, while throughput almost doubled.

Power reduction

To indirectly measure FPGA power use in the original untuned system, a current meter was added to the VCCint supply line, first testing VCCint set at the nominal 2.5 V, then with VCCint reduced to 2.2 V. Not unexpectedly, the power usage decreased by about 30 percent, but with only a 7.7 percent drop in time-related performance. This suggests a future application in which the operating system or application program could dynamically set a power budget. The hardware would then adjust VCC to use that power, and the TEAtime hardware would adapt the system to obtain the best performance under those conditions.

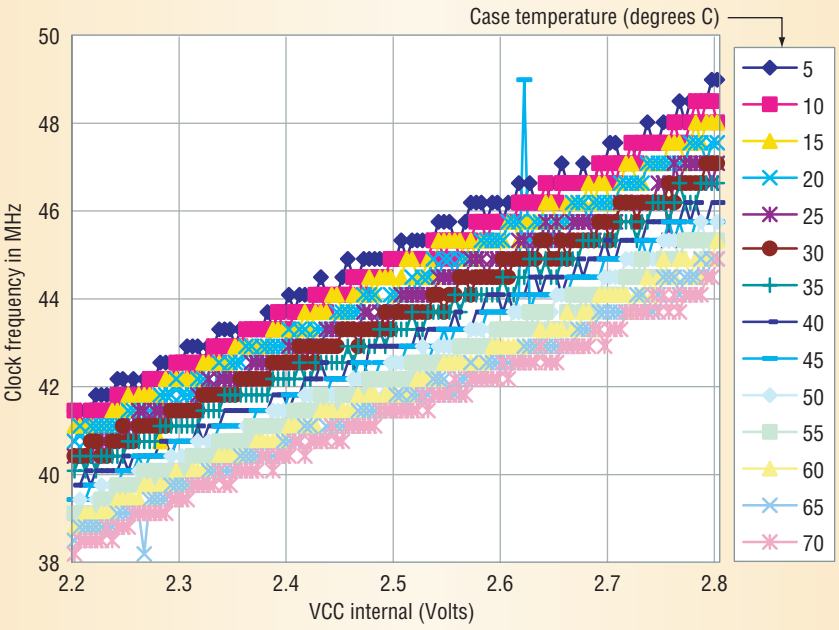


Figure 4. System clock frequency as a function of both supply voltage (VCCint) and case temperature. VCCint and temperature are both independent variables, while the clock frequency is a dependent variable. TEAtime successfully adapted to varying environmental and operating conditions—the two deviant points stem from frequency-measuring-circuit anomalies.

DESIGN CONSIDERATIONS

When implementing a TEAtime system, developers must consider several common digital system factors having unique TEAtime solutions. These factors include multiple worst-case paths and metastability—the unlikely state in which flip-flop output is unpredictable. There are also other factors that could be areas of concern but do not turn out to be, including the inductance-caused power-supply voltage-droop problem, created by large and rapid changes in power supply current—recall from physics: $V = L \times di/dt$.

Multiple worst-case paths

To design a target system for TEAtime use, and particularly to construct the tracking logic, the designer must determine the worst-case path in the system, then construct a one-bit-wide version of this logic and its wiring to mimic the worst-case delay. Next, the design must place the tracking logic as close to its corresponding real logic as possible—if not right in the middle of it—so that both components experience the same manufacturing, environmental, and operating conditions the real logic encounters.

This approach worked fine for the prototype, a very simple computer. However, complex microprocessor chips can have hundreds of worst-case paths, or worst-case paths within some small delta delay. These can all be different and exist in different operating environments because hot spots with varying temperatures and placements can occur on large chips.

To solve this problem, designers must construct the tracking logic for each possible worst-case path. Then, if any path indicates the clock frequency should be decreased, the DAC decrements. The DAC increments only if all tracking logic circuits indicate an increase.

This scenario raises yet another issue. The worst-case paths will likely be distributed throughout the chip. Given that it takes multiple clock cycles for signals to cross a chip, and more cycles are likely in the future with higher-speed clocks, the overall TEAtime control must be insensitive to such long delays.

The prototype only changed the operating frequency after every complete program execution, that is, after hundreds of cycles. The ideal TEAtime logic shown in Figure 1 actually had another flip-flop between the timing checker and the counter. This flip-flop was initially set, then cleared whenever the timing checker indicated a down signal. Over hundreds of cycles, if only one of the timing-checker samples indicated the clock frequency should be decreased, it was. This modification actually solves both the cross-chip delay problem—in which signals from the individual tracking logic circuits can use very low-speed paths—and the metastability issue.

Metastability

Flip-flop unpredictability occurs when both the data and clock inputs to a flip-flop change at or very close to the same time. In this case, the flip-flop can go into a metastable state in which its output is neither 0 nor 1: The output voltage level lies between the 0 and 1 thresholds. This means that circuits with inputs connected to the flip-flop will not only see a possibly incorrect value, but two different circuits could interpret this bogus value two different ways—one as 0, the other as 1—causing the system to malfunction.

Barring other stimulation, a metastable condition can last indefinitely. However, this is unlikely. Further, synchronous systems cannot avoid metastability completely—the best that developers

can do is minimize the likelihood it will occur and, if it does, minimize its duration.

In TEAtime, the timing checker and the feedback loop's basic construction increase the likelihood of a metastable condition. When the system is stable, the input to the timing checker always changes right about the time the system clock's rising edge occurs. This is not a problem with the modified circuit, which examines the timing checker's output over hundreds of cycles, because of the very low probability that conditions will exist long enough to create a lasting metastable condition. The TEAtime control loop's integral changes in clock frequency and the slight changes in clock timing—known as clock jitter, which is inherent in all synchronous systems—further reduce the likelihood of metastability.

di/dt and other adverse conditions

On large microprocessors, such as Intel's Itanium, millions of transistors can switch state simultaneously, leading to a big change in the power-supply current in a short time, also known as a large di/dt, or change (delta) in current per unit change in time. When combined with the inherent inductance of the power supply network both on and off chip, this results in large voltage spikes on the chip's power buses. These spikes are already as large as plus or minus 5 percent of the power supply voltage and could increase in future chip generations.[10] The tuned TEAtime prototype's current safety margin can handle the effects of even plus or minus 9 percent supply voltage spikes. Such spikes are an issue with any large synchronous system, not just TEAtime. TEAtime designers can handle this and other system reliability constraints by increasing the tracking logic's safety margin delay. This can be done either at design time or at runtime, the latter with or without software assistance.

Many synchronous systems today have multiple clock domains with different unsynchronized frequencies, such as Intel-style PCs with unsynchronized CPU and PCI clocks. Therefore, TEAtime's varying clock should not be an issue in production systems; it can be handled by existing design techniques.

Developers can now take advantage of typical operating conditions and improve synchronous-digital-system performance dramatically. Such adaptable systems are also excellent candidates for mobile and military applications, in which digital systems undergo exposure to extreme environmental conditions. ■

References

1. M. Olivieri, A. Trifiletti, and A. De Gloria, "A Low-Power Microcontroller with On-Chip Self-Tuning Digital Clock Generator for Variable-Load Applications," *Proc. 1999 Int'l Conf. Computer Design*, IEEE CS Press, 1999, pp. 476-481.
2. A.K. Uht, "Achieving Typical Delays in Synchronous Systems via Timing Error Toleration," tech. report 032000-0100, Dept. Electrical and Computer Eng., Univ. of Rhode Island, 10 Mar. 2000; www.ele.uri.edu/~uht.
3. D. Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *Proc. 2003 Int'l Symp. Microarchitecture*, IEEE CS Press, 2003, pp. 7-18.
4. A. Merchant et al., "Analysis of a Control Mechanism for a Variable Speed Processor," *IEEE Trans. Computers*, vol. 45, no. 7, 1996, pp. 968-976.
5. AMD Staff, "AMD PowerNow! Technology Brief," Advanced Micro Devices, Inc., 2002; www.amd.com/us-en/assets/content_type/DownloadableAssets/Power_Now2.pdf.
6. Intel Staff, "Intel Low-Power Technologies," Intel Corp., 2003; www.intel.com/ebusiness/products/related_mobile/wp021601_sum.htm?iid=ipp_battery+press_lowpow.
7. A.E. Sjogren and C.J. Myers, "Interfacing Synchronous and Asynchronous Modules within a High-Speed Pipeline," *Proc. 17th Conf. Advanced Research in VLSI* (ARVLSI 97), IEEE CS Press, 1997, pp. 47-61.
8. T. D. Burd et al., "A Dynamic Voltage Scaled Microprocessor System," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, 2000, pp. 1571-1580.
9. A.J. Martin, "Design of an Asynchronous Microprocessor," tech. report CS-TR-89-02, Computer Science Dept., California Inst. of Technology, 1989.
10. E. Grochowski, D. Ayers, and V. Tiwari, "Microarchitectural di/dt Control," *IEEE Design & Test of Computers*, May-June 2003, pp. 40-47.

Augustus K. Uht *is a research professor in the Department of Electrical and Computer Engineering, University of Rhode Island. His research interests include computer architecture, especially the field of instruction-level parallelism, focusing on the study and implementation of resource flow computing and disjoint eager execution to realize the execution of tens of instructions per cycle. Uht received a PhD in electrical engineering from Carnegie Mellon University. Contact him at uht@ele.uri.edu.*

Making Typical Silicon Matter with Razor

A codesign methodology incorporates timing speculation into a low-power microprocessor pipeline and shaves energy levels far below the point permitted by worst-case computation paths.

Todd Austin
David Blaauw
Trevor Mudge
University of Michigan

Krisztián Flautner
ARM Ltd.

An old adage says, "If you're not failing some of the time, you're not trying hard enough." To address the power challenges that current on-chip densities pose, we adapted this precept to circuit design. Razor,[1] a voltage-scaling technology based on dynamic detection and correction of circuit timing errors, permits design optimizations that tune the energy in a microprocessor pipeline to typical circuit operational levels. This eliminates the voltage margins that traditional worst-case design methodologies require and allows digital systems to run correctly and robustly at the edge of minimum power consumption. Occasional heavyweight computations may fail and require additional time and energy for recovery, but the overall computation in the optimized pipeline requires significantly less energy than traditional designs.

Razor supports timing speculation through a combination of architectural and circuit techniques, which we have implemented in a prototype Razor pipeline in 0.18-micrometer technology. Simulation results of the SPEC2000 benchmarks showed energy savings for every benchmark, up to a 64 percent savings with less than 3 percent performance impact for error recovery.

SPEED, ENERGY, AND VOLTAGE SCALING

Both circuit speed and energy dissipation depend on voltage.

The speed or clock frequency, f, of a digital circuit is proportional to the supply voltage, V_{dd}:

$$f \propto V_{dd}$$

The energy E necessary to operate a digital circuit for a time duration T is the sum of two energy components:

$$E = SCV_{dd}^2 + V_{dd}I_{leak}T$$

where the first term models the *dynamic power* lost from charging and discharging the capacitive loads within the circuit and the second term models the *static power* lost in passive leakage current—that is, the small amount of current that leaks through transistors even when they are turned off. The dynamic power loss depends on the total number of signal transitions, S, the total capacitance load of the circuit wire and gates, C, and the square of the supply voltage. The static power loss depends on the supply voltage, the rate of current leakage through the circuit, I_{leak}, and the duration of operation during which leakage occurs, T.

The dependence of both speed and energy dissipation on supply voltage creates a tension in circuit design: To make a system fast, the design must utilize high voltage levels, which increases energy demands; to make a system energy efficient, the design must utilize low voltage levels, which reduces circuit performance.

Dynamic voltage scaling has emerged as a powerful technique to reduce circuit energy demands. In a DVS system, the application or operating system identifies periods of low processor utilization that can tolerate reduced frequency. With reduced frequency, similar reductions are possible in the supply voltage. Since dynamic power scales quadratically with supply voltage, DVS technology can

0018-9162/04/$20.00 © 2004 IEEE

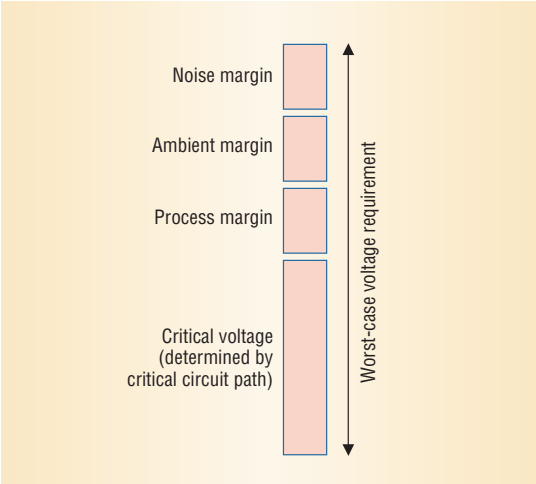


Figure 1. Critical voltage and margins to meet worst-case reliability requirements.

significantly reduce energy consumption with little impact on perceived system performance.[2]

ERROR-TOLERANT DVS

Razor is an error-tolerant DVS technology. Its error-tolerance mechanisms eliminate the need for voltage margins that designing for "always correct" circuit operations requires. The improbability of the worst-case conditions that drive traditional circuit design underlies the technology.

Voltage margins

Figure 1 shows margins for factors that can affect the voltage required to reliably operate a processor's underlying circuitry for a given frequency setting. First, of course, the voltage must be sufficiently high to fully evaluate the longest circuit computation path in a single clock cycle. Circuit designers typically use static circuit-level timing analysis to identify this *critical voltage*.

To the critical voltage, they add the following voltage margins to ensure that all circuits operate correctly even in the worst-case operating environment:

- *Process margins* ensure that performance uncertainties resulting from manufacturing variations in transistor dimensions and composition do not prevent slower devices from completing evaluation within a clock cycle. Designers find the margin necessary to accommodate slow devices by using pessimistically slow devices to evaluate the critical path's latency.
- *Ambient margins* accommodate slower circuit operations at high temperatures. The margin ensures correct operation at the worst-case temperature, which is typically 85-95°C.
- *Noise margins* safeguard against a variety of noise sources that introduce uncertainty in supply and signal voltage levels, such as di/dt noise in the supply voltage and cross-coupling noise in logic signals.

The sum of these voltages defines the minimum supply voltage that ensures correct circuit operation in even the most adverse conditions.

Worst-case improbability

In a simple experiment, we quantified the circuit error rates of an 18 × 18-bit multiplier block within a high-density field-programmable gate array. We used a Xilinx XC2V250-F456-5 FPGA because it contains full-custom multiplier blocks, which permit error-rate measurement with minimal routing-fabric overhead.

Test setup. Figure 2 shows the test harness and circuit schematic. The multiplier produces a 36-bit result each clock cycle. During FPGA logic placement, we directed synthesis to aggressively optimize the fast multiplier pipeline's performance. The resulting placement was fairly efficient; the Xilinx static timing analyzer indicated that 82 percent of the fast multiplier stage latency was in the custom multiplier block.

Each cycle, two 48-bit linear feedback shift registers (LFSRs) generate 18-bit uncorrelated random values, sending them to a fast multiplier pipeline and, in alternating cycles, to two slow multiplier pipelines. The slow pipelines take turns safely computing the fast pipeline's results, using a clock period that is twice as long as the fast multiplier pipeline.

As voltage decreases, values latched into the fast multiplier output latch may become *metastable*—that is, the values captured by the latch may be in transition between logic-0 and logic-1 and, thus, possess a voltage between these two well-defined values. The empty stage after the fast multiplier stage (labeled "Stabilize" in Figure 2) gives these potentially metastable values time to stabilize back to 0 or 1 before they are compared with the known-correct slow multiplier results.

A multiplexer on the output of the slow pipelines selects the correct result to compare with the fast pipeline's output. If the fast pipeline and slow pipeline results don't match, a circuit timing error has occurred, and the error counter is incremented.

We used the Xilinx static timing analyzer to evaluate the design's performance. The analyzer indicated that at 1.5 V and 85°C, the fast multiplier stage could run at up to 83.5 MHz; at 1.5 V and 27°C (room temperature), it could run at 88.6 MHz. All other support circuitry used to analyze multiplier errors was validated to 140 MHz. Thus, we are confident that all errors experienced in these experiments are localized to the fast multiplier pipeline circuits.

Error rates. Figure 3 illustrates the relationship between voltage and error rates for an 18 × 18-bit

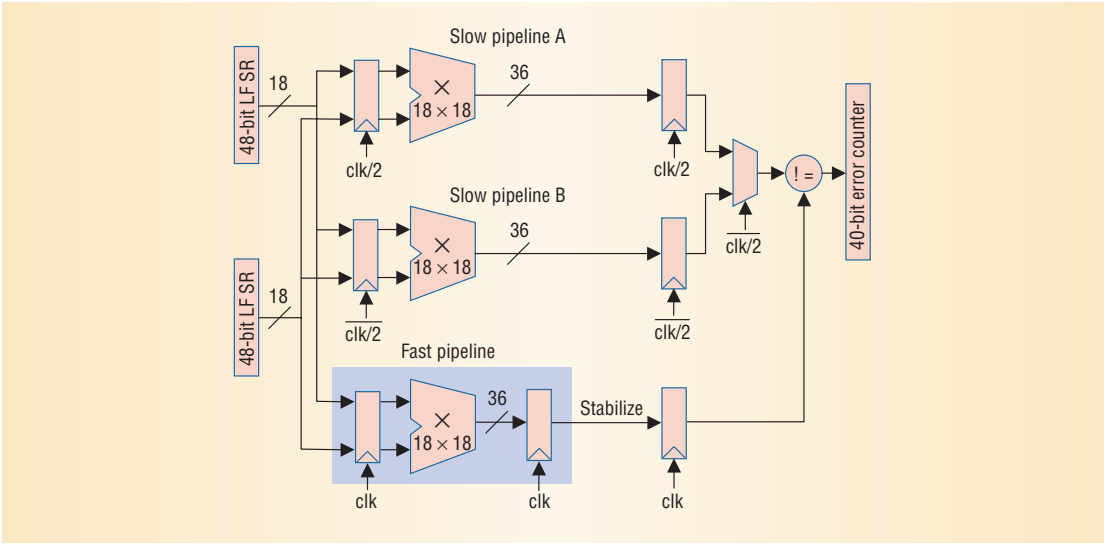


Figure 2. Error-rate test for 18 × 18-bit multiplier block. Shaded area in the test schematic indicates the multiplier circuit under test.

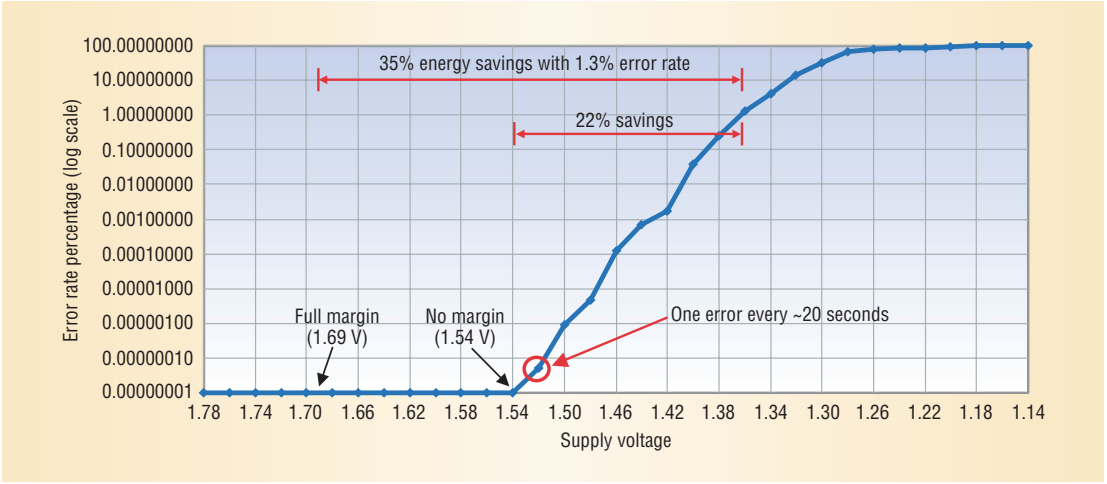


Figure 3. Measured error rates for an 18 × 18-bit FPGA multiplier block at 90 MHz and 27°C.

multiplier block running with random input vectors at 90 MHz and 27°C. The error rates are given as a percentage on a log scale.

The graph also shows two important design points:

- *no margin*—the lowest voltage that can still guarantee error-free circuit operation at 27°C, and
- *full margin*—the voltage at which the circuit runs without errors at 85°C in the presence of worst-case process variation and signal noise.

Traditional fault-avoidance design methodology sets the circuit voltage at the full margin point.

As Figure 3 shows, the multiplier circuit fails quite gracefully, taking nearly 180 mV to go from the point of the first error (1.54 V) to an error rate of 1.3 percent (1.36 V). At 1.52 V, the error rate is approximately one error every 20 seconds—or one error per 1.8 billion multiply operations.

The gradual rise in error rate is due to the dependence between circuit inputs and evaluation latency. Initially, only circuit inputs that require a complete critical-path reevaluation result in a timing error. As the voltage continues to drop, the number of internal multiplier circuit paths that cannot complete within the clock cycle increases, along with the error rate. Eventually, voltage drops to the point where none of the circuit paths can complete in the clock period, and the error rate reaches 100 percent.

Clearly, the worst-case conditions are highly improbable. The circuit under test experienced no errors until voltage has dropped 150 mV (1.54 V) below the full margin voltage. If a processor pipeline can tolerate a small rate of multiplier errors, it can operate with a much lower supply voltage. For instance, at 330 mV below the full margin voltage (1.36 V), the multiplier would complete 98.7 percent of all operations without error, for a total energy savings (excluding error recovery) of 35 percent.

RAZORED PROCESSOR ARCHITECTURE

Given the improbability of worst-case operating conditions, an opportunity exists to reduce voltage commensurate with typical operating conditions. The processor pipeline must, however, incorporate

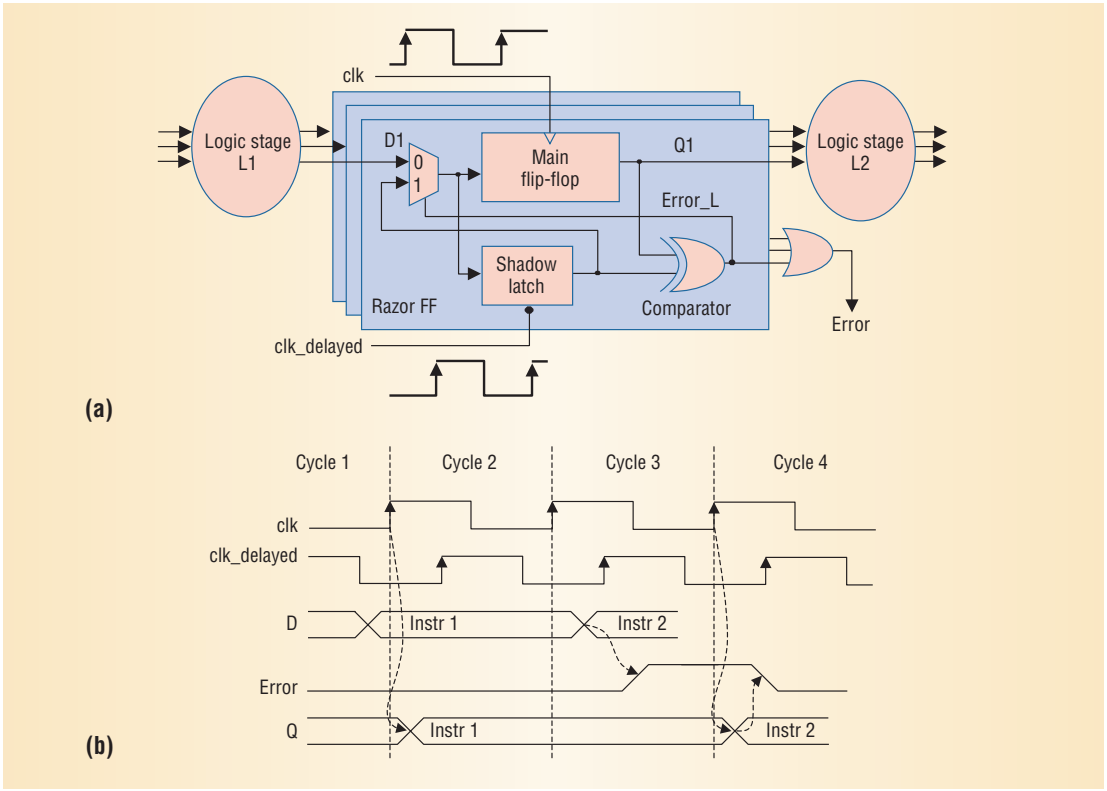


Figure 4. Razor flip-flop for a pipeline stage. (a) A shadow latch controlled by a delayed clock augments each flip-flop. (b) Razor flip-flop operation with a timing error in cycle 2 and recovery in cycle 4.

a timing-error detection and recovery mechanism to handle the rare cases that require a higher voltage. In addition, the system must include a voltage control system capable of responding to the operating condition changes, such as temperature, that might require higher or lower voltages.

Detecting circuit timing errors with Razor flip-flops

Figure 4a illustrates a Razor flip-flop for a pipeline stage. At the circuit level, a *shadow latch* augments each delay-critical flip-flop. A delayed clock controls the shadow latch.

Figure 4b illustrates a Razor flip-flop operation. In clock cycle 1, the combinational logic L1 meets the setup time by the clock's rising edge, and both the main flip-flop and the shadow latch will latch the same data. In this case, the error signal at the XOR gate's output remains low and the pipeline's operation is unaltered. In cycle 2, the combinational logic exceeds the intended delay due to subcritical voltage operation. In this case, the main flip-flop does not latch the data; but since the shadow latch operates using a delayed clock, it successfully latches the data in cycle 3.

To guarantee that the shadow latch will always latch the input data correctly, the allowable operating voltage is constrained at design time such that under worst-case conditions, the logic delay does not exceed the shadow latch's setup time. In cycle 3, a comparison of the valid shadow latch data with the main flip-flop data generates an error signal. In cycle 4, the shadow latch's data moves into the main flip-flop and becomes available to the next pipeline stage L2.

If a timing error occurs in a particular clock cycle of pipeline stage L1, the data in L2 in the following clock cycle is incorrect and must be flushed from the pipeline. However, since the shadow latch contains stage L1's correct output data, the pipeline does not need to reexecute the instruction through L1. This is a key feature of Razor: It reexecutes an instruction failure in one pipeline stage through the *following* stage, while incurring a one-cycle penalty. The proposed approach therefore guarantees an instruction's forward progress and avoids the perpetual reexecution of an instruction at a particular pipeline stage because of timing failure.

Razor flip-flop construction must minimize the power and delay overhead. The power overhead is inherently low because in most cycles a flip-flop's input will not transition; thus, the only power overhead incurred comes from switching the delayed clock. To minimize even this power requirement, Razor inverts the main clock to generate the delayed clock locally, thus reducing its routing capacitance.

Many noncritical flip-flops in a design will not need Razor technology. For example, if the maximum delay at a flip-flop input is guaranteed to meet the required cycle time under the worst-case subcritical voltage setting, it isn't necessary to replace it with a Razor flip-flop because it will never need to initiate timing recovery. In the prototype Razor pipeline designed to study this problem, for example, we found that only 192 of a total of 2,408 flip-flops required Razor.

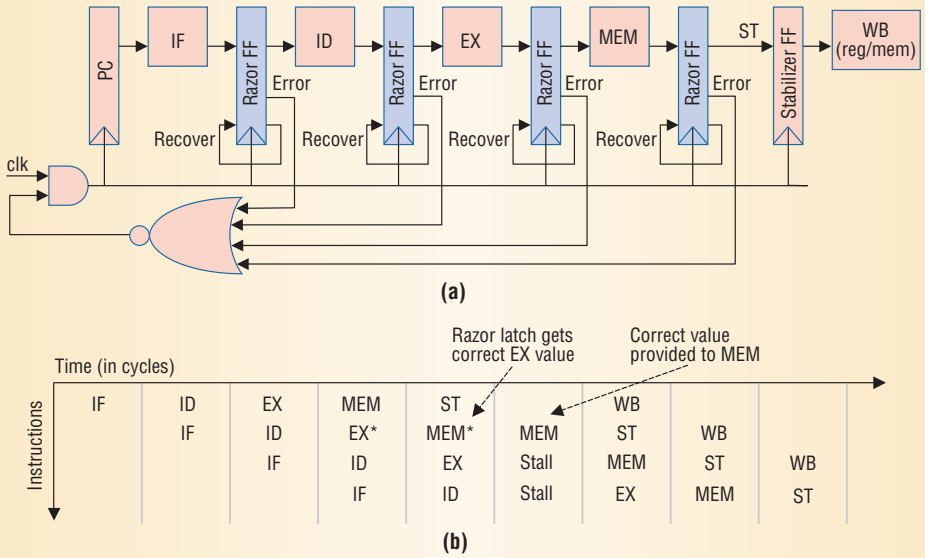


Figure 5. Pipeline recovery using global clock gating. (a) Pipeline organization and (b) pipeline timing for an error occurring in the execute (EX) stage. Asterisks denote a failing stage computation. IF = instruction fetch; ID = instruction decode; MEM = memory; WB = writeback.

Recovering pipeline state after timing-error detection

A pipeline recovery mechanism guarantees that any timing failures that do occur will not corrupt the register and memory state with an incorrect value. We have developed two approaches to recovering pipeline state.[1] The first is a simple method based on clock gating, while the second is a more scalable technique based on counterflow pipelining.[3]

Figure 5 illustrates pipeline recovery using a global clock-gating approach. In the event that any stage detects a timing error, pipeline control logic stalls the entire pipeline for one cycle by gating the next global clock edge. The additional clock period allows every stage to recompute its result using the Razor shadow latch as input. Consequently, recovery logic replaces any previously forwarded errant values with the correct value from the shadow latch.

Because all stages reevaluate their result with the Razor shadow latch input, a Razor flip-flop can tolerate any number of errant values in a single cycle and still guarantee forward progress. If all stages fail each cycle, the pipeline will continue to run but at half the normal speed.

In aggressively clocked designs, implementing global clock gating can significantly impact processor cycle time. Consequently, we have designed and implemented a fully pipelined recovery mechanism based on counterflow pipelining techniques. Figure 6 illustrates this approach, which places negligible timing constraints on the baseline pipeline design at the expense of extending pipeline recovery over a few cycles.

When a Razor flip-flop generates an error signal, pipeline recovery logic must take two specific actions. First, it generates a *bubble* signal to nullify the computation in the following stage. This signal indicates to the next and subsequent stages that the pipeline slot is empty. Second, recovery logic triggers the *flush train* by asserting the ID of the stage generating the error signal. In the following cycle, the Razor flip-flop injects the correct value from the shadow latch data back into the pipeline, allowing the errant instruction to continue with its correct inputs.

Additionally, the flush train begins propagating the failing stage's ID in the opposite direction of instructions. At each stage that the active flush train visits, a bubble replaces the pipeline stage. When the flush ID reaches the start of the pipeline, the flush control logic restarts the pipeline at the instruction *following* the failing instruction.

In the event that multiple stages generate error signals in the same cycle, all the stages will initiate recovery, but only the failing instruction closest to the end of the pipeline will complete. Later recovery sequences will flush earlier ones.

RAZOR PIPELINE PROTOTYPE

To obtain a realistic prediction of the power overhead for detecting and correcting circuit timing errors, we implemented Razor in a simplified 64-bit Alpha pipeline design, using Taiwan Semiconductor Manufacturing Co. 0.18-micrometer technology to produce the layout.[1] In addition to gate- and circuit-level power analysis on the error-detection-and-recovery design, we performed architectural simulations to analyze the overall throughput and power characteristics of Razor-based voltage reduction for different benchmark test programs. The benchmark studies demonstrated that, on average, Razor reduced simulated power consumption by nearly a factor of two—a greater than 40 percent reduction—compared to traditional design-time dynamic voltage scaling and delay chain-based approaches.

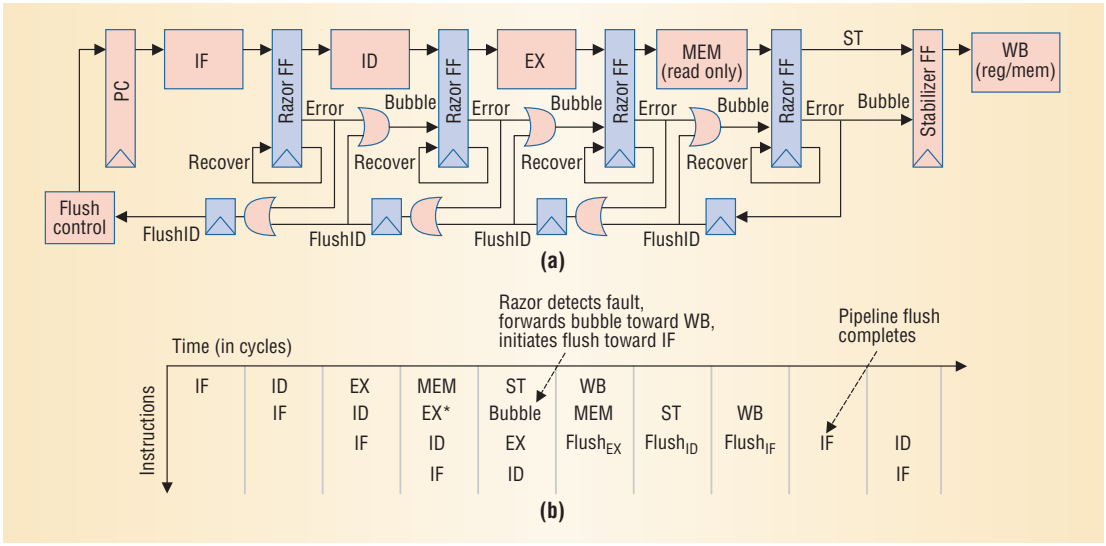


Figure 6. Pipeline recovery using counterflow pipelining. (a) Pipeline organization and (b) pipeline timing for an error occurring in the execute (EX) stage. Asterisks denote a failing stage computation. IF = instruction fetch; ID = instruction decode; MEM = memory; WB = writeback.

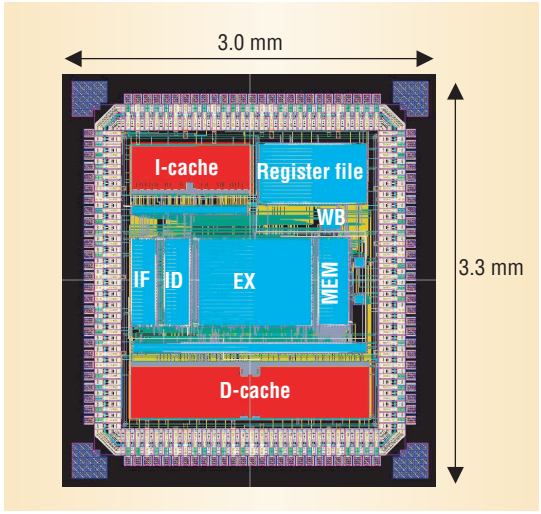


Figure 7. Razor prototype layout. A simplified 64-bit Alpha pipeline consists of instruction fetch, instruction decode, execute, and memory/writeback with both I- and D-cache.

Power analysis

Figure 7 shows the design layout; Table 1 lists the specifications and test results for error-free operation and for error-correction-and-recovery overhead. The pipeline consists of instruction fetch, instruction decode, execute, and memory/writeback with 8 Kbytes of both I-cache and D-cache. Performance analysis revealed that only the instruction decode and execute stages were critical at the worst-case voltage and frequency settings, thus requiring Razor flip-flops for their critical paths. While the overall design included a total of 2,408 flip-flops, only 192 of them implemented Razor technology. The clock for the Razor flip-flops was delayed by a half cycle from the system clock.

We performed both gate-level power simulations and SPICE (simulation program with integrated circuit emphasis) to evaluate the power overhead of the timing-failure detection and recovery circuit. The total power consumption during error-free operation at 200 MHz is 425 mW at 1.8 V.

Table 1 lists two energy consumption values, *switching* and *static*, for standard and Razor flip-flops over one clock cycle in error-free operation. These values reflect whether the latched data is changing or not changing, respectively. We expect a power overhead of 12.2 mW for inserting delay buffers to meet short-path constraints, bringing the total overhead for the detection and recovery circuitry in error-free operation to 3.1 percent of total power consumption.

The energy required to detect a setup violation, generate an error signal, and restore the correct shadow latch data into the main flip-flop was 210 femtojoules (10^{-15}) per such event for each Razor flip-flop. The total energy required to perform a single timing-error detection and recovery event in the pipeline was 189 picojoules (10^{-12}), resulting in an additional overhead of approximately 1 percent more total power when operating at a pessimistic 10 percent error rate.

Architectural benchmark tests

To further explore the design's efficiency, we developed an advanced simulation technique based on the SimpleScalar architectural tool set. This technique combines function-level architectural simulation with detailed SPICE-level circuit simulation, enabling the study of how voltage influences the timing of the pipeline stage computation.

Table 2 lists simulation results for the SPEC2000 benchmarks running on the simulated Razor prototype pipeline. Through extensive simulation, we identified the *fixed energy-optimal supply voltage* for each benchmark. This is the single voltage that results in the lowest overall energy requirement for each program. Table 2 also shows the average pipeline error rate, energy reduction, and pipeline throughput reduction (instructions per clock) at the fixed energy-optimal voltage. The total energy computation includes computation, Razor latch and check circuitry, and the total pipeline recovery energy incurred when an error is detected.

The Razor latches and error-detection circuitry

increase adder energy by about 4.3 percent. The energy for error detection and recovery is conservatively estimated at 18 times the cost of a single add (at 1.8 V), based on a six-cycle recovery sequence at typical activity rates.

Clearly, running the pipeline at a low error rate can reclaim significant energy. All of the benchmarks showed significant energy savings, ranging from 23.7 to 64.2 percent. One particularly encouraging result is that Razor mutes error rates and performance impacts up to and slightly past the energy-optimal voltage, after which the error rate rises very quickly. At the energy-optimal voltage point, the benchmarks suffered at most a 2.49 percent reduction in pipeline performance due to recovery flushes.

There appears to be little tradeoff in performance when fully exploiting energy savings at subcritical voltages. We have simulated voltages down to 0.6 V, but our Razor prototype design can only validate circuit timing down to 1.2 V. This constraint will limit the energy savings of four of the benchmarks.

Since additional voltage scaling headroom exists, we are examining techniques to further reduce voltage in future prototype designs.

Table 1. Razor prototype specifications.

Description	Specification
Technology node	0.18 mm
Voltage range	1.8 V to 1.2 V
Total number of logic gates	45,661
D-cache size	8 Kbytes
I-cache size	8 Kbytes
Die size	3 × 3 mm
Clock frequency	200 MHz
Clock delay	2.5 nS
Total number of flip-flops	2,408
Number of Razor flip-flops	192
Total number of delay buffers	2,498
Error-free operation	
Total power	425 mW
Standard FF energy (switching/static)	49 fJ / 95 fJ
Razor FF energy (switching/static)	60 fJ / 160 fJ
Total delay buffer power overhead	12.2 mW
Total power overhead	3.1%
Error correction and recovery overhead	
Energy per Razor FF per error event	210 fJ
Total energy per error event	189 pJ
Recovery power overhead at 10% error rate	1%

FUTURE WORK

We submitted the prototype Razor pipeline design for fabrication in October 2003 and expect to test the real silicon early this year.

Meanwhile, two immediate questions must be answered to fully implement Razor technology: How do we design razored control logic, and how do we design razored memories? Our initial research indicates that we can develop microarchitectural solutions to address delay failures that occur in the control logic, and we are investigating the use of double-sampling sense amplifiers for developing a Razor-enabled cache.

We see several potential applications of Razor technology in the future.

Self-tuning systems

In its current form, Razor sets voltage globally—chip-wide, but we could refine it to allow distributed voltage control. Under a distributed control system, each processor pipeline stage could operate at a separate, potentially different voltage deter-

Table 2. Energy-optimal characteristics for SPEC2000 benchmarks.

Program	Optimal V_{dd}	Error rate (percent)	Energy reduction (percent)	Pipeline throughput reduction (percent)
bzip	1.1	0.31	57.6	0.70
crafty	1.175	0.41	60.5	0.60
con	1.3	1.21	34.4	1.24
gap	1.275	1.15	30.1	2.49
gcc	1.375	1.62	23.7	1.47
gzip	1.3	1.03	35.6	0.41
mcf	1.175	0.67	48.7	0.00
parser	1.2	0.61	47.9	0.29
twolf	1.275	2.67	30.7	0.31
vortex	1.3	0.53	42.8	0.14
vpr	1.075	0.01	64.2	0.00
Average			42.4	

Meeting power challenges, especially for mobile systems, will require rule-breaking innovation.

mined by monitoring pipeline stage error rates. Releasing the constraint of a single operating voltage enables significant optimizations of voltage assignments across the processor stages, leading to further power savings.

Alternatively, we can maintain global voltage control but skew the clock phase individually for each unit to perform a type of dynamic retiming. High-clock-rate processors employ similar techniques to statically adjust clock skew.

Extreme voltage scaling

Current voltage-scalable designs are typically limited to operating voltages within 50 percent of maximum supply voltage.[4,5] This translates to a total power improvement of at most four times, due to the quadratic dependence of power on voltage.

However, our work on drowsy caches shows that memories can operate as low as the threshold voltage of their transistors (typically one-third of normal supply voltage).[6,7] In fact, it is possible to push the supply voltage to subthreshold levels as low as a few hundred millivolts. The power savings possible in such regimes is dramatic—approaching a factor of 10. The cost of bringing units out of drowsy mode when they are needed is an obstacle to this approach, but we have already solved many of the issues in this area. Operating at these levels introduces a degree of uncertainty in unit behaviors, which makes subthreshold voltage scaling an ideal application for Razor.

Reliability

Razor technology and extensions to it may help solve more general transient failures. For example, a number of radiation sources in nature can affect electronic circuit operations. The two most prevalent are

- *gamma rays*, which arrive from space (while the atmosphere filters out most of them, some occasionally reach the Earth's surface, especially at higher altitudes), and
- *alpha particles*, which are created when atomic impurities (found in all materials) decay.

When these energetic particles strike a very small transistor, they can deposit or remove sufficient charge to temporarily turn the device on or off, possibly creating a logic error.[8,9] They have posed a problem for dynamic RAM designs since the late 1970s when DRAM capacitors became sufficiently small to be affected by them.[10]

It is difficult to shield against natural radiation sources. Gamma rays that reach the Earth's surface have such high momentum that only thick, dense materials can stop them.[11] A thin shield can stop alpha particles, but if the shield is not free of atomic impurities, it becomes an additional source of natural radiation. Neither shielding approach is cost-effective for most system designs.

Furthermore, the smaller feature sizes that have driven the digital revolution make the particles relatively larger. Their impact (literally) is growing dramatically, and designers will likely be forced to adopt fault-tolerant design solutions to protect against them. Razor offers a solution that may compare well with conventional error-correcting codes.

Process variability

As feature sizes drop below 100 nanometers, the variation in key parameters, such as supply and threshold voltages and transistor widths, increases dramatically. These variations limit the guaranteed performance of circuits and potentially neutralize the benefits of smaller silicon geometries.

Razor removes the supply voltage design margins normally needed to account for worst-case technology variations between different chip instances (fabrication-time variability). Razor designs can also adjust dynamically to a computation's data-dependent nature, saving energy by lowering voltage when data dependencies permit (runtime variability).

Power is the next great challenge for computer systems designers, especially those building mobile systems with frugal energy budgets. We believe that meeting this challenge will require sustained rule-breaking innovation. Technologies like Razor enable "better than worst-case design," opening the door to methodologies that optimize for the common case rather than the worst.

Optimizing designs to meet the performance constraints of worst-case operating points requires enormous circuit effort, resulting in tremendous increases in logic complexity and device sizes. It is also power-inefficient because it expends tremendous resources on operating scenarios that seldom occur. Using recomputation to process rare, worst-case scenarios leaves designers free to optimize standard cells or functional units—at both their architectural and circuit levels—for the common case, saving both area and power. ■

Acknowledgments

This work was supported by ARM, an Intel Graduate Fellowship, the Defense Advanced Research Projects Agency, the Semiconductor Research Corporation, the Gigascale Silicon Research Center, and the National Science Foundation.

References

1. D. Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *Proc. 36th Ann. Int'l Symp. Microarchitecture* (MICRO-36), IEEE CS Press, 2003, pp. 7-18.
2. T. Mudge, "Power: A First-Class Architectural Design Constraint," *Computer*, Apr. 2001, pp. 52-58.
3. R.F. Sproull, I.E. Sutherland, and C.E. Molnar, "The Counterflow Pipeline Processor Architecture," *IEEE Design and Test of Computers*, Fall 1994, pp. 48-59.
4. K. Flautner and T. Mudge, "Vertigo: Automatic Performance-Setting for Linux," *Proc. 5th Conf. Operating Systems Design and Implementation* (OSDI), ACM Press, 2002, pp. 105-116.
5. K. Flautner et al., to appear in "IEM 926: An Energy-Efficient SoC with Dynamic Voltage Scaling," *Proc. Design Automation and Test in Europe* (DATE-2004), IEEE CS Press, 2004.
6. N. Kim et al., "Drowsy Instruction Caches: Leakage Power Reduction Using Dynamic Voltage Scaling and Cache Subbank Prediction," *Proc. 35th Ann. IEEE/ACM Symp. Microarchitecture* (MICRO-35), IEEE CS Press, 2002, pp. 219-230.
7. K. Flautner et al., "Drowsy Caches: Simple Techniques for Reducing Leakage Power," *Proc. 29th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, 2002, pp. 148-157.
8. J. Ziegler et al., "IBM Experiments in Soft Fails in Computer Electronics," *IBM J. Research and Development*, Jan. 1996, pp. 3-18.
9. P. Rubinfeld, "Managing Problems at High Speed," *Computer*, Jan. 1998, pp. 47-48.
10. T. May and M. Woods, "Alpha-Particle-Induced Soft Errors in Dynamic Memories," *IEEE Trans. Electron Devices*, vol. 26, no. 2, 1979, pp. 2-9.
11. J. Ziegler, "Terrestrial Cosmic Rays, *IBM J. Research and Development*, Jan. 1996, pp. 19-39.

Todd Austin *is an associate professor of electrical engineering and computer science at the University of Michigan. His research interests include computer architecture, compilers, computer system verification, and performance analysis tools and techniques. Austin received a PhD in computer science from the University of Wisconsin. Contact him at austin@umich.edu.*

David Blaauw *is an associate professor of electrical engineering and computer science at the University of Michigan. His research interests include VLSI design and CAD with emphasis on circuit analysis and optimization problems for high-performance and low-power microprocessor designs. Blaauw received a PhD in computer science from the University of Illinois, Urbana-Champaign. Contact him at blaauw@umich.edu or visit his Web site at http://blaauw.eecs.umich.edu.*

Trevor Mudge *is the Bredt Professor of Electrical Engineering and Computer Science at the University of Michigan. In addition, he runs Idiot Savants, a chip-design consultancy. His research interests include computer architecture, CAD, and compilers. Mudge received a PhD in computer science from the University of Illinois, Urbana-Champaign. He is a Fellow of the IEEE and a member of the ACM, the IEE, and the British Computer Society. Contact him at tnm@eecs.umich.edu.*

Krisztián Flautner *is the director of advanced research at ARM Ltd. and the architect of ARM's Intelligent Energy Management technology. His research interests include high-performance, low-power processing platforms to support advanced software environments. Flautner received a PhD in computer science and engineering from the University of Michigan. Contact him at krisztian.flautner@arm.com.*

10 great reasons to renew your IEEE Computer Society membership

1. New for 2004, an online reference book membership benefit – free!
2. Access to any or all of 100 distance-learning courses – free!
3. Personal subscription to *Computer* magazine – free!
4. Opportunity to subscribe to the complete IEEE Computer Society Digital Library or individual periodicals in your specialty area at the lowest available rates.
5. Advance notice of more than 150 IEEE Computer Society conferences, symposia, and workshops—plus generous discounts on registration fees.
6. Discounts on print books, tutorials, conference proceedings, and extended online reference book collections too!
7. Opportunities to participate in 40+ Technical Committees and over 160 Standards Working Groups.
8. Membership in the nearest of over 150 local chapters worldwide – free!
9. Prestigious email alias of your.name@computer.org – free!
10. Be part of the profession and a network of over 100,000 of the best and brightest computing professionals around the world.

IEEE

IEEE COMPUTER SOCIETY

Do it today!

www.ieee.org/renewal

Speeding Up Processing with Approximation Circuits

Approximation can increase a microprocessor's clock frequency by replacing a complete logic function with a simplified circuit that mimics the function and uses rough calculations to speculate and predict results.

Shih-Lien Lu
Intel

Microprocessor performance has accelerated rapidly in recent years, primarily by achieving gains on two fronts. On one front, microarchitecture innovations have taken advantage of the increasing number of devices to process more useful instructions per cycle, predominantly through the superscalar[1] approach. On the other front, device miniaturization improves layout density and makes the circuits run faster because electrons travel a shorter distance.

A superscalar processor issues multiple instructions and executes them with multiple identical function units. It employs dynamic scheduling techniques and executes instructions outside the original program's order. Superscalar processing seeks mainly to exploit as much instruction-level parallelism as possible in the program.

Clever new circuit techniques further accelerate the logic as well. Together with finer pipe stages, its greatly accelerated clock frequency gives the modern microprocessor more cycles per unit time.

The combination of these two advances lets the system process more instructions per unit of time. However, most researchers believe that to continue exploitation of larger instruction-level parallelism, even more complexity will be necessary.[2]

This complexity increase tends to cause more circuit delay in the pipeline's critical path, thus limiting the clock frequency from rising further. The current approach lets logic structures with long delays spread over multiple pipe stages, which causes logic structures that completed the computation in single pipe stages previously to take more than one cycle time.

Using finer pipeline stages increases pipeline latencies and imposes higher penalties due to branch misprediction and other misspeculation. Moreover, other instructions that depend on the results of these multistaged functional blocks must wait until they finish to move forward in the pipeline.

Some researchers[3] have demonstrated the impact of data dependencies and branch penalties on pipeline performance. Since these two factors converge to undercut the performance gain of increasing pipeline stages, there is an optimal number of pipe stages with which a microprocessor will achieve maximum performance. This means that increasing pipe stage and frequency alone does not guarantee improved performance. Therefore, these long-delaying logic structures could become the microprocessor performance bottleneck as clock frequency continues to rise.

Thus, one challenge in achieving higher performance in future microprocessors is to increase instructions per cycle without compromising the increase in clock frequency. Current microprocessors use circuits based on Boolean functions as elementary units. The processor employs a global timing reference to synchronize data transfer between units.

For this type of synchronous system, it is essential to know the maximum time needed to compute a function, also known as the worst-case circuit delay. The depth of a circuit in gates, as well as each gate's delay, determines the worst-case delay's upper

0018-9162/04/$20.00 © 2004 IEEE

Approximation uses rough calculations to speculate and predict results.

bound, which decides the frequency of the global timing reference. However, it is well known that a unit's delay depends strongly on its input. Usually, a circuit finishes the computation earlier than the worst-case delay—this interval is referred to as the *typical delay*. In these cases, the system still must wait for the maximum time bound to guarantee that the result is correct every time.

Faced with these constraints, my colleagues and I have turned to *approximation* to increase the clock's frequency. In this approach, instead of implementing the complete logic function necessary to realize a desired functionality, a simplified circuit mimics it. In contrast to traditional value prediction,[4] which relies on value history or value behavior, approximation uses rough calculations to speculate and predict results. The approximation circuit usually produces the correct result. If the approximate circuit fails—which usually occurs in worst-case delays—the machine employing approximation circuits must recover. Recovery degrades the overall performance. Thus, it is essential to ensure that the gain from reducing the worst-case delay outweighs the recovery overhead.

CRITICAL PIPELINE STAGES

We have applied the approximation concept to a few stages in a superscalar processor: execution, rename logic, and issue logic.

Execution

This stage consists of an approximate adder and a Booth's multiplier.

Approximate adder. Many instructions require addition. Load, store, and branch use the adder for address calculation. Arithmetic instructions use the adder for add, subtract, multiply, and divide calculations. The adder is a key performance structure in function units.

There are many different kinds of adders. Due to performance requirements, most current high-performance processors employ one of the known parallel adders. These parallel adders, such as carry-look-ahead, Brent-Kung, Kogge-Stone, and carry-select, all have comparable asymptotic performance when implemented in CMOS with either static or dynamic circuits. That is, their critical path delay is asymptotically proportional to log (N), where N is the number of bits used in the addition. The cost complexity of parallel adders approaches $N\log N$ when the fan-in and fan-out of the gates used are fixed.

The adder's full carry chain determines the critical path. To generate the correct final result, the calculation must consider all input bits to obtain the final carry out. However, in real programs, inputs to the adder are not completely random and the effective carry chain is much shorter for most cases. Thus, it is possible to build a faster adder with a much shorter carry chain to approximate the result. Since carry chains are usually much shorter, a design that considers only the previous k inputs (look-ahead k-bits) instead of all previous input bits for the current carry bit can approximate the result:

$c_i = f(a_{i-1}, b_{i-1}, a_{i-2}, b_{i-2}, \ldots, a_{i-k}, b_{i-k})$, where $0 < k < i + 1$ and $a_j, b_j = 0$ if $j < 0$.

Given that the delay cost of calculating the full carry chain length of N bits is proportional to log (N), if k equals the square root of N, the new approximation adder will only need on the order of half the delay. With random inputs, the probability of having a correct result considering only k previous inputs is:

$$P(N,k) = \left(1 - \frac{1}{2^{k+2}}\right)^{N-k-1}.$$

This is derived with the following steps. First consider why the prediction is incorrect. If we only consider k previous bits to generate the carry, the result will be wrong only if the carry propagation chain is greater than $k + 1$. Moreover, the previous bit must be in the carry-generate condition.

This can only happen with a probability of $1/2^{k+2}$ if we consider a k-segment. Thus, the probability of being correct is one minus the probability of being wrong. Second, there are a total of $N - (k + 1)$ segments in an N-bit addition. To produce the final correct result, the segment should not have an error condition. We multiply all the probabilities to produce the final product. This equation could determine the risk taken by selecting the value of k. For example, assuming random input data, a 64-bit approximation adder with 8-bit look-ahead ($k = 8$) produces a correct result 95 percent of the time.

Figure 1 shows a sample approximation adder design with $k = 4$. The top and bottom rows are the usual carry, propagate, and generate circuits. The figure also shows the sum circuits used in other parallel adders. However, the design implements the carry chain with 29 4-bit carry blocks and three boundary cells. These boundary cells are similar but smaller in size. A Manchester carry chain could implement 4-bit carry blocks. Thus, the critical path

delay is asymptotically proportional to constant with this design, and the cost complexity approaches N. In comparison with Kogge-Stone or Han-Carlson adders, this design is faster and smaller.

Since we know exactly what causes a result to be incorrect, the design could—and probably should—implement an error indication circuit. Whenever a carry propagation chain longer than k bits occurs, the approximation adder circuit will give an incorrect result. That is, for the ith carry bit, if the logic function (a_{i-1} XOR b_{i-1}) AND (a_{i-2} XOR b_{i-2}) AND ... AND (a_{i-k} XOR b_{i-k}) AND (a_{i-k-1} AND b_{i-k-1}) is true, the prediction will be wrong. The adder could implement this logic function for each carry bit and perform the logical OR of all these $n - 4$ outputs to signal if the approximation is incorrect. Instead of comparing the result of a fully implemented adder with this approximated adder, the error indication circuit provides a signal to select the correct result earlier in the process to help the pipeline recover.

Booth multiplier. Current microprocessors use Booth encoding and the Wallace tree to perform the multiply function. For radix-8 Booth encoding, generating $3x$ lies in the critical path. Unlike $2x$, $4x$—which can be generated by shifting—there is no easy way to generate $3x$ besides performing the actual addition. Again, the adder could approximate the multiplier to accelerate the frequency or reduce the number of cycles required. The straightforward approach uses the approximation adder to generate $3x$. However, its accuracy is unsatisfactory for the overall multiplication because errors accumulate. Remembering that an error occurs only when a carry propagation chain stretches longer than k, inspecting the differences between the correct and approximated result reveals that the discrepancy between them is always an N-bit vector sparsely populated with 1's separated by at least k 0's. Writing the equation using the propagate and generate (p,g) terms gives:

$$p_{i-1} \text{ AND } p_{i-2} \text{ AND } \ldots \text{ AND } p_{i-k} \text{ AND } g_{i-k-1}.$$

To obtain the correct multiplication result, these error vectors must be included in the Wallace tree. A simpler circuit also can approximate the sum of error vectors. For example, an OR gate can approximate the summation of all error vectors. This adds only one extra partial product for the Wallace tree to compress. The added delay to the tree is negligible, but the process has reduced the delay to generate $3x$. Developers also could use this method to design a nonapproximated multiplier.

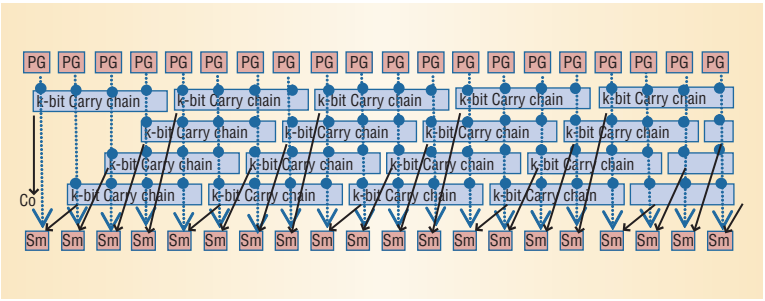


Figure 1. Sample 32-bit approximation adder. The adder has the usual carry, propagate, generate, and sum circuits but also implements a carry chain with 29 4-bit carry blocks and three boundary cells.

Rename logic

The register rename logic's critical path centers on the associative lookup delay and the priority logic when multiple matches are found. Experiments with benchmarks revealed that dependent instructions can have spatial locality: The instructions are most likely to be close to each other. Thus, the design can use a smaller *content-addressable memory* to implement the mapping table.

The CAM table basically contains a portion of the entire map. When a new instruction enters the rename logic, the CAM renames its destination binding and assigns it a new physical binding. The mapping table then updates if it is not full. Otherwise, the processor drops the oldest binding to leave room for the newly renamed destination binding.

At the same time, the processor uses the source bindings to look up the partial CAM. If it finds no physical mapping in the small CAM, but the mapping does exist in the full CAM, a misspeculation occurs. Since the number of inputs to the priority encoder equals the number of entries in the smaller CAM, the rename logic delay is also smaller.

Using a much smaller CAM table that contains only a number of instructions equal to the latest N-square-root register mapping, where N is the window size, doubles the speed. Given the locality property of register dependency, most of the reading operation from the rename logic should be correct. In addition to the faster approximation renaming logic, this design retains a regular CAM and the associated full-length priority encoder. This will recover the misspeculation and provide the correct renaming result in the next cycle.

Issue logic

This approach uses this same idea by targeting the issue logic on the earliest square-root-N entries so that the issue logic only needs to consider waking up, selecting, and bypassing data to instructions within square-root-N entries to the head of register update unit (RUU). Because resistance-capacitance dominates the wakeup and bypass delay and RC delay is more sensitive to window size, the speed increase will more than double in these two

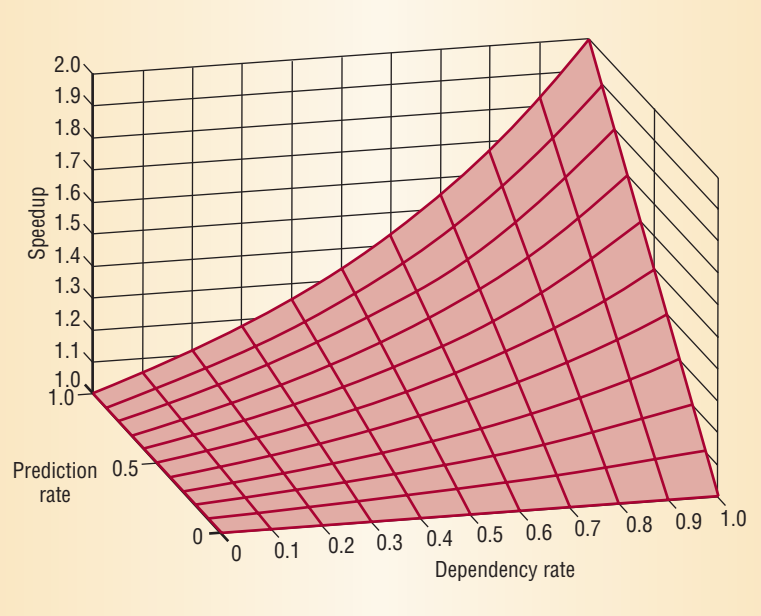


Figure 2. Speedup of speculative execution as a function of prediction rate and instruction dependency rate. The functional unit writeback-bandwidth-occupancy rate is 50 percent.

logics. Thus, the total speculative issue logic delay will be less than half the issue logic in the baseline microarchitecture if only square-root-*N* entries are considered. The approximated issue logic does not need a replay because this approach generates no false results. However, some bandwidth or functional units may be wasted because the square-root-*N* entries might not have enough ready instructions.

PERFORMANCE IMPACT

Using analytical modeling and simulation helps to assess the optimization's impact on performance.

Simple analytical modeling

Philip Emma and Edward Davidson[3] showed that as the length of any pipeline increases, data dependencies and branches monotonically degrade the pipeline performance in terms of clock cycles per instruction. The longer the pipeline, the more penalty cycles the data dependency and branch misprediction cause. However, increasing the pipeline length increases clock frequency monotonically. These two opposing factors will decide the optimal pipeline length based on specific technology.

A simple analytical model based on an in-order machine overcomes these data dependencies. The baseline model and speculative model run at the same frequency. In the baseline machine, the functional unit has a two-cycle execution time and the speculative unit has a one-cycle execution time with a replay penalty. Obviously, under the same frequency, the model with the shorter pipeline will suffer less from data dependency and branch mispredictions. However, the model will replay the wrongly speculated result, which will occupy more functional-unit-writeback bus bandwidth, reducing the performance gain.

The following are major factors in the performance comparison:

- prediction rate (PR) of the speculative logic,
- data-dependency rate (DR) for the instructions,
- functional unit writeback bus occupancy rate (FR), and
- overall branch miss rate (BR).

Notice that the overall branch miss rate is the product of the branch miss rate and branch frequency. Since the goal is to evaluate data dependency, the comparison simplifies the branch prediction factor to one term.

Since both machines modeled have the same frequency, the comparison performance metric is mainly cycle per instruction. The formula used for CPI with data dependency and branch penalty is

$$CPI = 1 + PD \times C_{Dstall} + PB \times C_{Bstall}$$

where PD is the probability that data dependency exists between two adjacent instructions, PB is the probability that an arbitrarily selected instruction is a branch and that the branch prediction is wrong. C_{Dstall} and C_{Bstall} are the corresponding stalled cycle when the dependency and branch misprediction takes place.

For simplicity, the comparison further assumes that C_{Bstall} is three cycles for both models. For the baseline pipe stage structure, PD is data dependency rate DR, and the stalled cycle is 1. Thus:

$$CPI = 1 + DR + PB \times C_{Bstall}.$$

A pipeline structure with speculative functions has four boundary cases. Either all instructions are

1. independent and the prediction rate is 100 percent,
2. independent and the prediction rate is 0,
3. dependent and the prediction rate is 100 percent, or
4. dependent and the prediction rate is 0.

For cases 1 and 2, when the prediction is perfect, there is no data-dependency penalty. For case 2, the verification logic will reissue the instruction in the next cycle. This requires an extra writeback slot for the reissued instruction. While the impact of the extra writeback slot on performance is complicated, the comparison can approximate the rela-

tionship by the following method: If the functional unit's FR is 100 percent for the original instructions, the extra writeback will always stall the pipeline by one cycle. If the original instructions have an occupancy rate of 50 percent or less, this extra writeback will not stall the pipeline. CPI will not be affected in this case.

A linear approximation interpolates the relationship between C_{Dstall} and FR. The linear equation needs to satisfy the two boundary conditions mentioned:

$$C_{Dstall}(FR) = 2 \times FR - 1.$$

For case 4, the analysis resembles case 2 but differs in that all instructions are dependent so that the pipeline will be stalled for one cycle even when no limitation on writeback bandwidth exists. This means that the pipeline will be one cycle when FR is 50 percent and two cycles when FR is 100 percent:

$$C_{Dstall}(FR) = 2 \times FR.$$

Combining all cases together with branch prediction provides the following:

$$CPI = 1 + (2 \times FR - 1) \times (1 - DR) \times (1 - PR) + 2 \times FR \times DR \times (1 - PR) + PB \times C_{Bstall}.$$

Let the speedup be the ratio of baseline CPI and speculative CPI. Figures 2, 3, and 4 show the speedup when FR is 0.5, 0.8, and 0.95, respectively. These figures assume either no branch misprediction impact or a perfect branch prediction rate. As the diagrams indicate, the speedup rate increases monotonically with dependency rate and prediction rate.

When the functional-unit occupation rate is high, the speculative method is more likely to sacrifice performance because replay instructions cause more penalties in writeback bandwidth. If FR is more than 50 percent, when the prediction rate or dependency rate is low enough, the speculative microarchitecture performance drops below the baseline. In an extreme case, when the dependency rate is 0, the speedup increases with prediction rate. However, the maximum speedup rate is 1, which means no speedup occurs even with perfect prediction and, with less than perfect prediction, performance actually decreases. The results show that the speculative method only works for the case in which the instruction-dependency rate is significant.

In another extreme case, when the prediction rate is 0, the speedup increases with the dependency rate but remains lower than 1. This means that the speculation method requires a minimum prediction rate to achieve any performance improvement.

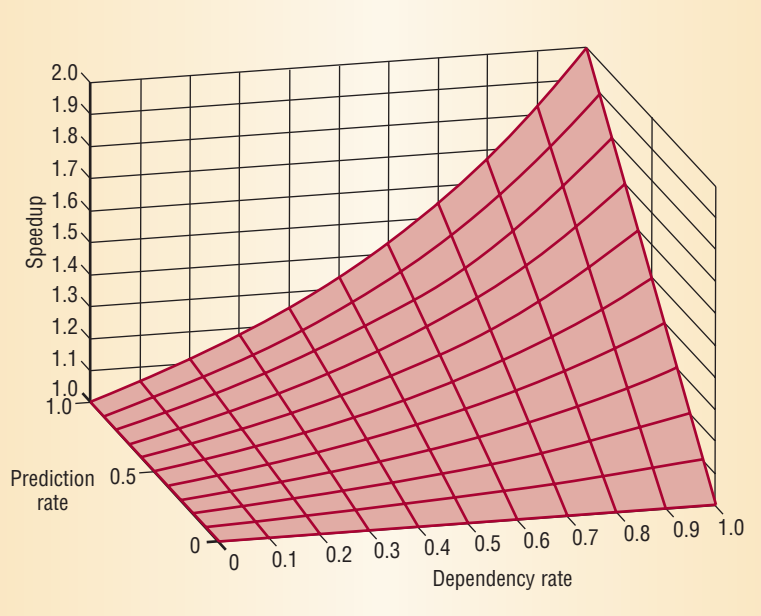


Figure 3. Speedup of speculative execution as a function of prediction rate and instruction-dependency rate. The functional unit writeback-bandwidth-occupancy rate is 80 percent.

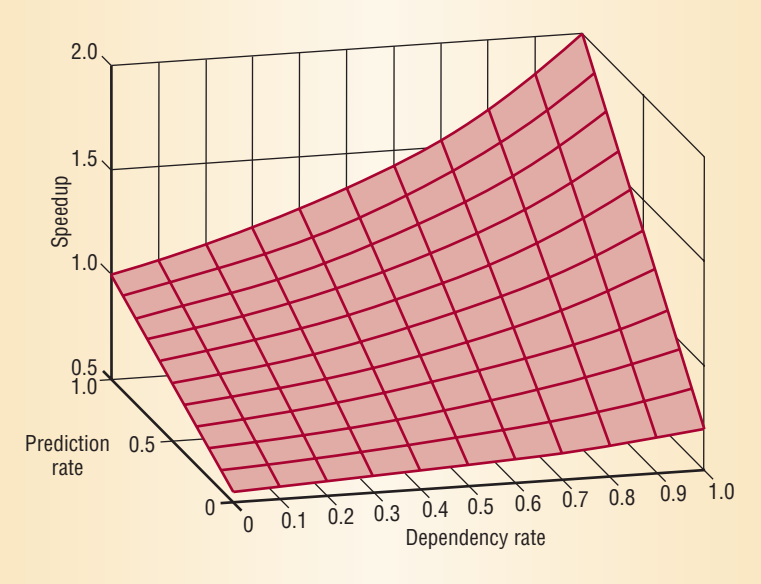


Figure 4. Speedup of speculative execution as a function of prediction rate and instruction-dependency rate. The functional unit writeback-bandwidth-occupancy rate is 95 percent.

Figures 5 and 6 show the impact of the overall branch misprediction rate and dependency rate on performance when the data prediction rate is high and the writeback-occupancy rate is medium. At the lower data-dependency rate, when speculative performance is low, the performance speedup increases as the PB increases; at the higher data-dependency rate, when speculative speedup is high, the speedup decreases as the PB increases.

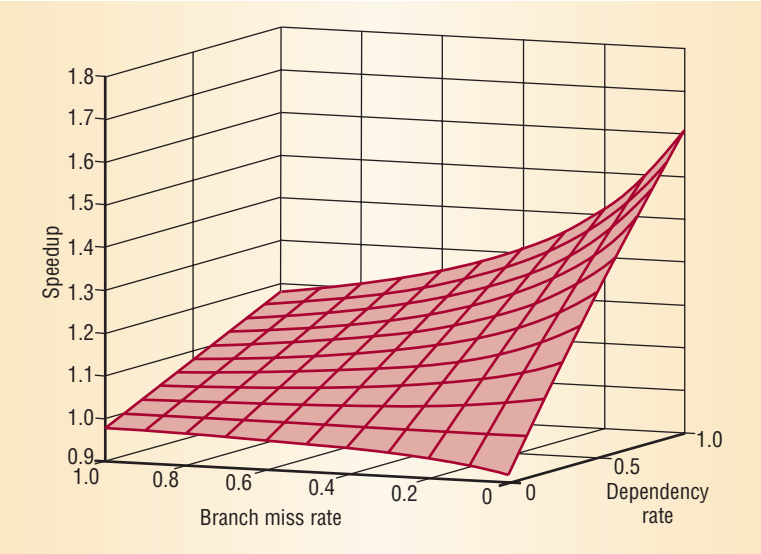


Figure 5. Speedup of speculative execution as a function of overall branch-misprediction rate and instruction-dependency rate. The functional unit writeback bandwidth occupancy rate is 80 percent and the data-prediction rate is 85 percent.

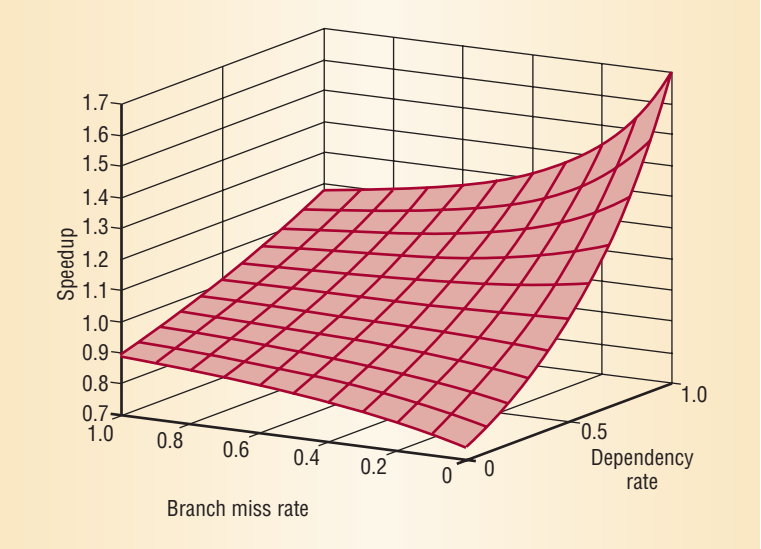


Figure 6. Speedup of speculative execution as a function of overall branch-misprediction rate and instruction-dependency rate. The functional unit writeback bandwidth occupancy rate is 80 percent and the data-dependency rate is 70 percent.

In the first case, data speculation suffers more in replay penalties than it gains in speculation performance because this case lacks dependent instructions. In the second case, the baseline model suffers more from a dependency stall penalty than it gains in performance from not replaying wrongly speculated instructions. The higher PB also causes more performance loss on both models at the same rate, thus it reduces the performance ratio.

The overall branch-misprediction rate favors the worst-case model because it compromises the data-dependency rate's performance impact. Figure 6 shows the relationship of speedup to PB and PR. For the same reason, the overall branch-misprediction rate will compromise the data-prediction rate's impact. This analysis suggests that a good branch-prediction rate is important for reaping the benefits of data speculation.

Simulation

The SimpleScalar tool set[5] provides a method for comparing the speculative microarchitecture's performance with the baseline machine.

Method. Assume both models run with the same frequency. In the baseline machine, to maintain the frequency, all cycle-limiting logic blocks take two cycles. In the speculative machine with approximation circuits, these same logic blocks take only one cycle. However, the speculative machine will need to replay when it incorrectly generates the result and incurs a misspeculation penalty. The independent simulation experiment uses the rename logic, issue logic, and adder, assuming that only one of these components provided the main performance limiter. The simulation, which ran several benchmarks from the SPEC suite, used the reference input.

Findings. These experiments revealed that the approximation adder's accuracy is much higher than the derived probability using random data inputs. With random data, the anticipated prediction accuracy was around 65 percent for 32-bit addition with a 4-bit carry chain. However, the simulation results show that close to 90 percent of the addition is correct using the approximation adder with a 4-bit carry and inputs from real applications.

The approximated rename logic produces close to 80 percent correct results. However, the approximated issue window logic has a low accuracy of only about 40 percent. The experiments evaluated the impact of approximation variance in two parameters: issue width and out-of-order window size.

After setting the RUU window size to 64, issue width to 4, integer adder number to 4, and integer multiplier number to 1, the experiment ran two billion instructions for each benchmark. Next, the study shuffled the parameters to obtain a window size of 16×32, an issue width of 8, an integer adder number of 8, and an integer multiplier number of 2, and ran each benchmark for 500 million instructions. Then the performance differences were compared with the baseline machine, normalized to one. The simulation showed that using approximation to speculate data as described does improve overall performance. For adder speculation, the performance improvement is less than the other two speculations.

The simulation achieved these results because addition completes close to the machine's back end. It is thus more likely to pollute the dependent instructions by false writebacks, thereby inducing more penalties. By reducing window size, the adder speculation performance relative to the baseline machine increases. A smaller issue window has fewer independent instructions, which produces a higher dependency between instructions and helps execution complete more quickly.

On the other hand, increasing issue width and the number of function units degrades relative performance. Wider issue width, larger window size, and more functional units potentially produce more instruction-level parallelism. More instructions execute per cycle. Every time a misspeculation occurs, the penalty becomes larger, outweighing the performance gain from resolving the dependency chain earlier.

For rename and issue logics, the simulation also reduces the speculative window size so that the worst-case delay is close to half the worst-case delay of the baseline window. This will compromise the relationship between relative performance and window size, issue width, and functional unit.

When the issue width is 8, with 64 entry-instruction windows and eight execution units—which represent a wide-issue machine with large window size and many more functional units—the relative performance of programs with inherent instruction-level parallelism suffers. For example, the performance of ijpeg degrades significantly with issue speculation.

With this application, the issue-speculation prediction accuracy of 24 percent—caused by the reduced issue window size—is low. This means that the number of ready instructions in the approximation window is much smaller than the real number of ready instructions. Even though approximating issue window size exacts no recovery penalty, the loss of parallelism from approximation degrades performance greatly. It also causes a huge waste of execution bandwidth, as the analytical model validates.

Programs with high instruction-level parallelism have a lower data-dependency rate. Programs with a lower data-dependency rate will not benefit much from shortening the pipeline length. Combined with the low prediction accuracy that a smaller window size causes, pipeline performance suffered when using approximation.[6]

Using an approximation circuit to reduce the number of cycles a function requires is the first step in achieving variable pipeline delays based on data values.[7] An asynchronous pipeline such as the micropipeline that Ivan Sutherland proposed[8] will provide the ultimate variable delay due to data values. Each stage of an asynchronous pipeline will take only as much time as necessary to process the data. Once it completes the evaluation, the pipeline can forward the result to the next stage for processing without waiting for the worst-case delay. Although a misspeculation penalty does not occur, signaling the completion of execution at each stage does cause handshaking overhead.

So far, there is still no efficient way to detect a stage's completion. Perhaps other means can be found to extend the approximation concept so that it can simplify more microprocessor stages to take advantage of the typical delay. ■

References

1. J.E. Smith and G.S. Sohi, "The Microarchitecture of Superscalar Processors," *Proc. IEEE*, vol. 83, no. 12, 1995, pp. 1609-1624.
2. S. Palacharla et al., "Complexity-Effective Superscalar Processors," *Proc. 24th ACM/IEEE Int'l Symp. Computer Architecture*, IEEE Press, 1997, pp. 206-218.
3. P.G. Emma and E.S. Davidson, "Characterization of Branch and Data Dependencies in Programs for Evaluating Pipeline Performance," *IEEE Trans. Computers*, vol. 36, no. 7, 1987, pp. 859-876.
4. M.H. Lipasti and J.P. Shen, "Exceeding the Dataflow Limit via Value Prediction," *Proc. 29th IEEE/ACM Int'l Symp. Microarchitecture*, IEEE Press, 1996, pp. 226 -237.
5. D.C. Burger and T.M. Austin, "The SimpleScalar Tool Set, v. 2.0," tech. report 1342, Computer Science Dept., Univ. of Wisconsin, Madison, 1997.
6. T. Liu and S-L. Lu, "Performance Improvement with Circuit-Level Speculation," *Proc. 33rd ACM/IEEE Int'l Symp. Microarchitecture*, IEEE Press, 2000, pp. 348-355.
7. S. Komori et al., "An Elastic Pipeline Mechanism by Self-Timed Circuits," *IEEE J. Solid-State Circuits*, vol. 23, no. 1, 1988, pp. 111-117.
8. I.E. Sutherland, "Micropipelines," *Comm. ACM*, vol. 32, no. 6, 1989, pp. 720-738.

Shih-Lien Lu *is a senior staff researcher at Intel's Hillsboro Microarchitecture Lab. His research interests include computer microarchitecture, embedded systems, self-timed circuits, and VLSI systems design. Lu received a PhD in computer science from the University of California, Los Angeles. He is a member of the IEEE Computer Society and the IEEE. Contact him at shih-lien.l.lu@intel.com.*

Not A Member Yet?

Here Is What You're Missing...

Distance Learning

Members of the IEEE Computer Society enjoy FREE access to a comprehensive distance learning program for computer professionals. Join today and take any or all of 100 online course titles through our Distance Learning Campus. Subjects include Java, project management, Cisco networks, UNIX, Windows, XML, Oracle, SQL, and more.

Magazines and Journals

A FREE subscription to *Computer* magazine is included with your membership. You may also subscribe to other publications in your area of expertise at member discounts. Or subscribe to the *IEEE Computer Society Digital Library*—21 periodicals and 1200+ conference proceedings—for a complete online resource.

Conferences and Workshops

Enhance your knowledge and share practical experiences at more than 150 conferences, workshops, and symposia held each year worldwide. Members save at least 25% on registration fees and get advance notice of the meetings.

Other Valuable Benefits

- FREE membership in your local chapter
- FREE e-mail alias of Your.Name@computer.org
- FREE membership in up to four of 40+ Technical Committees
- FREE membership in 160+ Standards Working Groups
- Member discounts on hundreds of books and conference proceedings

Join the IEEE Computer Society Today!

Complete the adjacent membership application today. For fastest service, apply online now at

http://computer.org/join

IEEE

IEEE COMPUTER SOCIETY

2004 IEEE Computer Society Professional Membership/Subscription Application

Membership and periodical subscriptions are annualized to and expire on 31 December 2004.
Pay full or half-year rate depending upon the date of receipt by the IEEE Computer Society as indicated below.

Membership Options*

All prices are quoted in U.S. dollars

	FULL YEAR Applications received 16 Aug 03 - 29 Feb 04	HALF YEAR Applications received 1 Mar 04 - 15 Aug 04
1 I do not belong to the IEEE, and I want to join **just** the Computer Society	$ 99 ❑	$50 ❑
2 I want to join **both** the Computer Society and the IEEE:		
I reside in the United States	$189 ❑	$95 ❑
I reside in Canada	$170 ❑	$85 ❑
I reside in Africa/Europe/Middle East	$166 ❑	$83 ❑
I reside in Latin America	$159 ❑	$80 ❑
I reside in Asia/Pacific	$160 ❑	$80 ❑
3 I already belong to the IEEE, and I want to join the Computer Society. (IEEE members need only furnish name, address, and IEEE number with payment.)	$ 42 ❑	$21 ❑

Are you now or were you ever a member of the IEEE?
Yes ❑ No ❑ If yes, provide member number if known: _____

Add Periodicals**

	ISSUES PER YEAR	FULL YEAR Applications received 16 Aug 03 - 29 Feb 04 PRINT	ELECTRONIC	COMBO	HALF YEAR Applications received 1 Mar 04 - 15 Aug 04 PRINT	ELECTRONIC	COMBO
IEEE Computer Society Digital Library **BEST DEAL**	n/a	n/a	$109 ❑	n/a	n/a	$55 ❑	n/a
Computing in Science and Engineering	6	$42 ❑	$34 ❑	$55 ❑	$21 ❑	$17 ❑	$28 ❑
IEEE Computer Graphics and Applications	6	$39 ❑	$31 ❑	$51 ❑	$20 ❑	$16 ❑	$26 ❑
IEEE Design & Test of Computers	6	$37 ❑	$30 ❑	$48 ❑	$19 ❑	$15 ❑	$24 ❑
IEEE Intelligent Systems	6	$37 ❑	$30 ❑	$48 ❑	$19 ❑	$15 ❑	$24 ❑
IEEE Internet Computing	6	$39 ❑	$31 ❑	$51 ❑	$20 ❑	$16 ❑	$26 ❑
IT Professional	6	$40 ❑	$32 ❑	$52 ❑	$20 ❑	$16 ❑	$26 ❑
IEEE Micro	6	$37 ❑	$30 ❑	$48 ❑	$19 ❑	$15 ❑	$24 ❑
IEEE MultiMedia	4	$35 ❑	$28 ❑	$46 ❑	$18 ❑	$14 ❑	$23 ❑
IEEE Pervasive Computing	4	$41 ❑	$33 ❑	$53 ❑	$21 ❑	$17 ❑	$27 ❑
IEEE Security & Privacy	6	$41 ❑	$33 ❑	$53 ❑	$21 ❑	$17 ❑	$27 ❑
IEEE Software	6	$44 ❑	$35 ❑	$57 ❑	$22 ❑	$18 ❑	$29 ❑
IEEE/ACM Transactions on Computational Biology and Bioinformatics **NEW**	4	$35 ❑	$28 ❑	$46 ❑	$18 ❑	$14 ❑	$23 ❑
IEEE/ACM Transactions on Networking†	6	$44 ❑	$33 ❑	$55 ❑	$22 ❑	$17 ❑	$28 ❑
IEEE Transactions on:							
Computers	12	$41 ❑	$33 ❑	$53 ❑	$21 ❑	$17 ❑	$27 ❑
Dependable and Secure Computing **NEW**	4	$31 ❑	$25 ❑	$40 ❑	$16 ❑	$13 ❑	$20 ❑
Information Technology in Biomedicine†	4	$35 ❑	$25 ❑	$44 ❑	$18 ❑	n/a	$22 ❑
Knowledge and Data Engineering	12	$43 ❑	$34 ❑	$56 ❑	$22 ❑	$17 ❑	$28 ❑
Mobile Computing	4	$30 ❑	$24 ❑	$39 ❑	$15 ❑	$12 ❑	$20 ❑
Multimedia†	6	n/a	n/a	$38 ❑	n/a	n/a	n/a
NanoBioscience†	4	$30 ❑	$24 ❑	$38 ❑	$15 ❑	n/a	$19 ❑
Parallel and Distributed Systems	12	$40 ❑	$32 ❑	$52 ❑	$20 ❑	$16 ❑	$26 ❑
Pattern Analysis and Machine Intelligence	12	$44 ❑	$35 ❑	$57 ❑	$22 ❑	$18 ❑	$29 ❑
Software Engineering	12	$38 ❑	$30 ❑	$49 ❑	$19 ❑	$15 ❑	$25 ❑
Visualization and Computer Graphics	6	$34 ❑	$27 ❑	$44 ❑	$17 ❑	$14 ❑	$22 ❑
VLSI Systems†	12	n/a	n/a	$28 ❑	n/a	n/a	$14 ❑
IEEE Annals of the History of Computing	4	$31 ❑	$25 ❑	$40 ❑	$16 ❑	$13 ❑	$20 ❑

Choose PRINT for paper issues delivered via normal postal channels.
Choose ELECTRONIC for 2004 online access to all issues published from 1988 forward.
Choose COMBO for both print and electronic.

Payment Information

Payment required with application

Membership fee $ _____
Periodicals total $ _____
Applicable sales tax*** $ _____
Total $ _____

Enclosed:
❑ Check/Money Order****

Charge my:
❑ MasterCard ❑ Visa
❑ American Express ❑ Diner's Club

_____
Card number

_____
Expiration date (month/year)

_____
Signature

USA-only include 5-digit billing zip code
☐☐☐☐☐

* Member dues include $17 for a 12-month subscription to *Computer*.

** Periodicals purchased at member prices are for the member's personal use only.

*** Canadian residents add 15% HST or 7% GST to total. AL, AZ, CO, DC, NM, and WV add sales tax to all periodicals. GA, IN, KY, MD, and MO add sales tax to print and combo periodicals. NY add sales tax to electronic periodicals. European Union residents add VAT tax to electronic periodicals.

**** Payable to the IEEE in U.S. dollars drawn on a U.S. bank account. Please include member name and number (if known) on your check.

† Not part of the IEEE Computer Society Digital Library. Electronic access is through www.ieee.org/ieeexplore.

For fastest service, apply online at http://computer.org/join

NOTE: In order for us to process your application, you must complete and return BOTH sides of this form to the office nearest you:

Asia/Pacific Office
IEEE Computer Society
Watanabe Bldg.
1-4-2 Minami-Aoyama
Minato-ku, Tokyo 107-0062 Japan
Phone: +81 3 3408 3118
Fax: +81 3 3408 3553
E-mail: tokyo.ofc@computer.org

Publications Office
IEEE Computer Society
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA 90720-1314 USA
Phone: +1 714 821 8380
Fax: +1 714 821 4641
E-mail: help@computer.org

Allow up to 8 weeks to complete application processing. Allow a minimum of 6 to 10 weeks for delivery of print periodicals.

IF4C

Personal Information

Enter your name as you want it to appear on correspondence.
As a key identifier in our database, circle your last/surname.

Male ❑ Female ❑ Date of birth (Day/Month/Year) _____

Title | First name | Middle | Last/Surname

Home address

City | State/Province

Postal code | Country

Home telephone | Home facsimile

Preferred e-mail

Send mail to: ❑ Home address ❑ Business address

Educational Information

First professional degree completed | Month/Year degree received

Program major/course of study

College/University | State/Province | Country

Highest technical degree received | Program/Course of study

Month/Year received

College/University | State/Province | Country

Business/Professional Information

Title/Position

Years in current position | Years of practice since graduation

Employer name | Department/Division

Street address | City | State/Province

Postal code | Country

Office phone | Office facsimile

I hereby make application for Computer Society and/or IEEE membership and agree to be governed by IEEE's Constitution, Bylaws, Statements of Policies and Procedures, and Code of Ethics. I authorize release of information related to this application to determine my qualifications for membership.

Signature | Date

APPLICATION MUST BE SIGNED

NOTE: In order for us to process your application, you must complete and return both sides of this form.

IEEE

IEEE COMPUTER SOCIETY

BPA Information

This information is used by society magazines to verify their annual circulation. Please refer to the audit codes and indicate your selections in the box provided.

A. Primary line of business

1. Computers
2. Computer peripheral equipment
3. Software
4. Office and business machines
5. Test, measurement and instrumentation equipment
6. Communications systems and equipment
7. Navigation and guidance systems and equipment
8. Consumer electronics/appliances
9. Industrial equipment, controls and systems
10. ICs and microprocessors
11. Semiconductors, components, sub-assemblies, materials and supplies
12. Aircraft, missiles, space and ground support equipment
13. Oceanography and support equipment
14. Medical electronic equipment
15. OEM incorporating electronics in their end product (not elsewhere classified)
16. Independent and university research, test and design laboratories and consultants (not connected with a manufacturing company)
17. Government agencies and armed forces
18. Companies using and/or incorporating any electronic products in their manufacturing, processing, research, or development activities
19. Telecommunications services, telephone (including cellular)
20. Broadcast services (TV, cable, radio)
21. Transportation services (airlines, railroads, etc.)
22. Computer and communications and data processing services
23. Power production, generation, transmission, and distribution
24. Other commercial users of electrical, electronic equipment and services (not elsewhere classified)
25. Distributor (reseller, wholesaler, retailer)
26. University, college/other education institutions, libraries
27. Retired
28. Others (allied to this field) _____

B. Principal job function

1. General and corporate management
2. Engineering management
3. Project engineering management
4. Research and development management
5. Design engineering management - analog
6. Design engineering management - digital
7. Research and development engineering
8. Design/development engineering - analog
9. Design/development engineering - digital
10. Hardware engineering
11. Software design/development
12. Computer science
13. Science/physics/mathematics
14. Engineering (not elsewhere classified)
15. Marketing/sales/purchasing
16. Consulting
17. Education/teaching
18. Retired
19. Other _____

C. Principal responsibility

1. Engineering or scientific management
2. Management other than engineering
3. Engineering design
4. Engineering
5. Software: science/management/engineering
6. Education/teaching
7. Consulting
8. Retired
9. Other _____

D. Title

1. Chairman of the Board/President/CEO
2. Owner/Partner
3. General Manager
4. V.P. Operations
5. V.P. Engineering/Director Engineering
6. Chief Engineer/Chief Scientist
7. Engineering Manager
8. Scientific Manager
9. Member of Technical Staff
10. Design Engineering Manager
11. Design Engineer
12. Hardware Engineer
13. Software Engineer
14. Computer Scientist
15. Dean/Professor/Instructor
16. Consultant
17. Retired
18. Other Professional/Technical _____

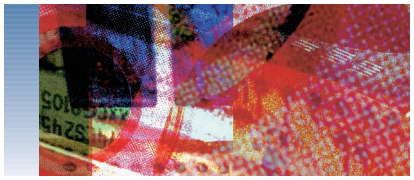
CSIDC 2004 Attracts Hundreds of Student Teams

More than 240 teams from 145 colleges and universities around the world have entered the first phase of the Fifth Annual Computer Society International Design Competition. The CSIDC strives to promote excellence in education by having undergraduate student teams design and implement computer-based solutions to real-world problems. The competition emphasizes teamwork in the design, implementation, and testing of a computer-based system. Interest in the competition has grown steadily since its inception in 2000: The first year of the challenge included only 50 competitor teams, selected from 155 interested schools.

Only one team per university can compete in the next phase of the CSDIC. To meet this requirement, schools will host their own competitions to decide which group of three to four students will make the cut for the international challenge. The current roster of competitor schools is listed on the following page.

In keeping with the CSIDC 2004 theme of "Making the World a Safer Place," teams are working to design projects that address global issues of safety, security, and reliability. Last year's winner, National Taiwan University, reflected the 2003 theme of "Added Value" with their "NEWS—Novel Educative Wireless Style," a tablet PC-based classroom system.

Project reports are due to CSIDC judges by 23 April and, based on the progress described in the reports, the top 10 teams will be announced on 24 May. Those finalists will present their prototype devices in a live World Finals event, set for 27-29 June in Washington, D.C. Teams that advance



to the Finals will compete for a $15,000 first-place team prize and a $10,000 grant to the sponsoring school that may be used to support instruction and research in computer science and engineering. The second-place team will receive $10,000, and the third-place team $6,000. All other teams in the top ten will receive $2,000.

In addition to the cash prizes, each student on the final ten teams will be given a one-year complimentary membership in both the Computer Society and the IEEE and a one-year subscrip-

Computer Society Seeks Three Awards Nominations by 31 July

The IEEE Computer Society is seeking nominations for three of its most prestigious awards: the Seymour Cray Computer Science and Engineering Award, the Sidney Fernbach Memorial Award, and the Computer Science and Engineering Undergraduate Teaching Award.

The Cray Award recognizes individuals whose innovative contributions to high-performance computing systems best exemplify the creative spirit demonstrated by supercomputing pioneer Seymour Cray. Recipients of the Cray Award receive a crystal memento, an illuminated certificate, and a $10,000 honorarium. Recent Cray honorees include Glen Culler, John Hennessy, and Monty Denneau.

The Computer Society established the Sidney Fernbach Memorial Award to recognize excellence in high-performance computing applications and to memorialize a pioneer in the application of high-performance computers to solve large computational problems. Conference committees associated with SC 2004 will evaluate nominees for the award. The Fernbach award winner will receive a certificate and $2,000.

A special awards ceremony at the SC 2004 conference will honor winners of the Cray and the Fernbach awards.

The Computer Society each year bestows the Undergraduate Teaching Award in Computer Science and Engineering on a professor or group of professors who best exemplify a commitment to undergraduate education through both teaching and service. The award can also acknowledge professors' efforts to increase the visibility of the Society. Honorees receive a plaque and a $2,000 honorarium.

IEEE Computer Society awards recognize technical achievements, contributions to engineering education, and service to the Society or the profession. Nominations for the Cray, Fernbach, and Undergraduate Teaching Awards are due by **31 July**. Most other Society awards have a 31 October deadline. Nomination materials for all Computer Society awards are available at www.computer.org/awards/.

tion to a Computer Society magazine.

In addition to the main awards, teams that place in the top 10 at CSIDC 2004 will be eligible for two special prizes. The Microsoft Multimedia Award is presented to the team whose presentation at the World Finals makes the most interesting, innovative, exciting, and appropriate use of multimedia. The Microsoft Award for Software Engineering recognizes the project that best exemplifies the application of good software engineering principles to the design and testing of a device prototype.

CSIDC is sponsored by Microsoft, with additional support from ABB and the IEEE Foundation. ■

CSIDC Participating Teams for 2004

Ain Shams University (2 teams)
Allama Iqbal Open University
American University of Beirut (3 teams)
Aristotle University of Thessaloniki
Banaras Hindu University
Beijing Broadcasting University (2 teams)
Beijing University of Technology (6 teams)
Belorussian State University
Bharathiar University
Binghamton University
Boston College
Boston University (2 teams)
Bradley University
Buffalo State College
Cairo University (6 teams)
California State Polytechnic University, Pomona (2 teams)
California State University, Long Beach
California State University, Chico
California State University, Northridge
Carleton University (2 teams)
Carnegie Mellon University
Carthage College
Catholic University of Colombia
Champlain College (2 teams)
Chiang Mai University
Clarion University of Pennsylvania
Concordia University
Cooper Union for the Advancement of Science and Art (2 teams)
DePaul University (2 teams)
DeVry Institute of Technology, Long Island (2 teams)
DeVry University, Phoenix
Don Bosco Technical College (4 teams)
Eastern Mediterranean University
Erciyes University
Florida Atlantic University
Florida Gulf Coast University
Fr. Conceicao Rodrigues College of Engineering (4 teams)
George Brown College
Georgia Southern University
Georgia State University
GI Khan Institute of Engineering Sciences and Technology (3 teams)
Grand Valley State University (2 teams)
Hampton University (2 teams)
Harbin Institute of Technology, Weihai
Hashemite University
Haverford College
Hindustan College of Engineering
Humboldt University, Berlin
Indian Institute of Technology, Bombay (3 teams)
Indian Institute of Technology, Kanpur (6 teams)
Indian Institute of Technology, Madras
Indian Institute of Technology, Roorkee
Indian Institute of Technology, Delhi
Indira Gandhi Institute of Technology
Industrial University of Santander (3 teams)
Interamerican University of Puerto Rico
International Institute of Information Technology (2 teams)
Iowa State University (2 teams)
James Madison University
Javeriana Pontifical University (3 teams)
Jawaharlal Nehru Engineering College
K.J. Somaiya Institute of Engineering and Information Technology
Kettering University
Kharkiv National University of Radioelectronics
Kuang Wu Institute of Technology
Lahore University of Management Sciences (4 teams)
Lebanese American University
Louisiana State University (2 teams)
Michigan Technological University
Midwestern State University
Military Institute of Engineering (5 teams)
Monroe County Community College
Nanjing University
Nanyang Technological University (12 teams)
National Institute of Science and Technology
National Taiwan University
National Technological University, Córdoba
National University of Sciences and Technology, Punjab
North Carolina State University
Northeastern University
Northern Virginia Community College
Ohio Northern University
Oregon State University
Pennsylvania State University, Erie
Polytechnical University of Bucharest
Polytechnic University, Brooklyn
Poznan University of Technology (2 teams)
Purdue University, Calumet
San Diego State University
San Jose State University (2 teams)
Santa Clara University (2 teams)
Sathyabama Institute of Science and Technology
School of Planning and Architecture, New Delhi
Seoul National University
Sir Syed University of Engineering and Technology (2 teams)
Slovak University of Technology
Southern Polytechnic State University
Sri Vidyanikethan Engineering College
Sri Venkateswara College of Engineering (3 teams)
State University of New York, Stony

Brook (4 teams)
State University of New York, Potsdam
Technical University of Iasi (2 teams)
Technological Institute of Aeronautics
Thadomal Shahani Engineering College
Thiagarajar College of Engineering
Tribhuvan University
United States Military Academy, West Point
University of Akron
University of Belgrade
University of Bridgeport
University of British Columbia
University of Central Oklahoma
University of Connecticut
University of Florida
University of Guelph (2 teams)
University of Karlsruhe
University of Kocaeli
University of Massachusetts, Dartmouth
University of Nebraska, Omaha
University of Nebraska, Lincoln (5 teams)
University of New South Wales (4 teams)
University of Novi Sad
University of Podlasie (2 teams)
University of Pretoria (2 teams)
University of São Paulo
University of Saskatchewan
University of Siena
University of South Carolina (2 teams)
University of South Florida
University of Suceava
University of Virginia
University of Wisconsin, Madison
Utah State University
Vasavi College of Engineering (7 teams)
Vidya Pratishthan's Institute of Information Technology
Virginia Polytechnic Institute
Vivekanand Education Society Institute of Technology (15 teams)
West Bengal University of Technology
Xavier University
Yildiz Technical University
Zagazig University

Register Now for Software Developer Credential

Software developers with two or more years of experience can apply now to take the IEEE Computer Society Certified Software Development Professional test. The CSDP credential provides well-qualified individuals an objective recognition of their superior abilities and commitment to lifelong learning in software development.

In a certification program unique in the software engineering field, the Computer Society evaluates individuals for CSDP status. Practitioners who are certified as Software Development Professionals enjoy a number of benefits, ranging from exposure to significant engineering theory, to gains in employment distinction and career confidence.

Any experienced software engineer interested in receiving external validation of his or her software engineering skills can test for CSDP certification. The CSDP test focuses on project management, systems architecture, and software practices from a customer satisfaction point of view.

SOFTWARE ENGINEERING STANDARDS

As the field of software engineering expanded in the 1990s, numerous technical certification programs began to emerge. Many were driven by applica-

Editor: Bob Ward, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; bnward@computer.org

Merwin Student Scholarship Applications Due by 31 May

The IEEE Computer Society encourages active members of its student branches to apply for the 2004-2005 Richard E. Merwin Student Scholarship. The scholarship, created in honor of a past president of the society, recognizes leaders in Computer Society student branch chapters who show promise in their academic and professional efforts.

Up to four scholarships of $3,000 each are available, paid in three quarterly installments starting in September.

Active members of a Computer Society student branch chapter who are juniors, seniors, or graduate students in electrical or computer engineering, computer science, or a computer-related field of engineering are eligible to apply. Applicants must be full-time students and are required to have a minimum 2.5 GPA.

Other awards and scholarships offered to students by the Computer Society include the Lance Stafford Larson best paper contest and the Upsilon Pi Epsilon/Computer Society Award for Academic Excellence, which is administered jointly by the IEEE Computer Society and the Upsilon Pi Epsilon international honor society.

For more information on Computer Society student scholarships and awards, visit www.computer.org/students/schlrshp.htm. Merwin Scholarship applications are due by **31 May**.

tion-specific requirements, as is the case with Novell, Microsoft, or Linux certification. Others were driven by project- or occupation-specific requirements. The IEEE Computer Society saw a need for one broad, objective certification program that recognized a level of advanced skill and knowledge in software development.

In contrast to what is sometimes characterized as "code-and-fix" programming, IEEE Standard 610.12 defines software engineering as "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software." With this in mind, Society leaders developed the CSDP as a standards-based certification. The CSDP credential, the only worldwide certification program to incorporate exam-based evaluation, relevant professional experience, and mandatory continuing education into its requirements, is part of the Society's larger effort to provide education and certification services to the software engineering community. The program builds on the Society's 20-year history in software engineering standards development.

TEST DETAILS

The CSDP examination consists of 180 multiple-choice questions based on concepts that should be familiar to software engineers with six or more years of experience. Test developers defined the subject matter based on a formal task analysis of the knowledge and skills necessary for successful job performance. The examination includes questions from 11 topic areas, including software design, software testing, and software requirements.

CSDP candidates must hold a baccalaureate degree and must have at least two years of software engineering experience within the four-year period prior to the application. Candidates must also have a total of at least 9,000 hours of relevant experience.

CSDP examinations are administered by Prometric, which performs live, computer-based testing at hundreds of locations throughout the world. The CSDP exam is offered at locations in the US, Canada, Brazil, China, Hungary, India, Ireland, Japan, and Russia.

Applications for the Spring 2004 testing window, which is open from 1 April to 30 June, are due by **1 April**. For the Fall 2004 testing window, which is open from 1 September to 30 October, applications are due by **15 August**. CSDP application and examination fees are $400 for IEEE or Computer Society members and $500 for nonmembers. Two to three weeks after an application is accepted, approved candidates will be mailed an *authorization to test*. Candidates must receive the authorization before scheduling an appointment to take the exam. Further application information is available at www.computer.org/certification/apply.htm.

SPECIAL TESTING AND PREP COURSE OPPORTUNITIES

To help candidates prepare for the CSDP exam, the Computer Society has invited author Richard Thayer to conduct a CSDP training class at the 2004 Systems and Software Technology Conference. The original teacher and developer of training material for the CSDP, Thayer's involvement in the Society's certification efforts earned him an honorary CSDP credential in the program's inaugural year. After Thayer's course, a paper and pencil exam will be administered on 23 April.

SSTC 2004 takes place in Salt Lake City from 17 to 19 April. For information on conference and course registration, visit www.stc-online.org.

Candidates may register for the CSDP exam at SSTC by completing the CSDP application form and submitting it by **7 April**. In addition to this deadline extension, candidates taking the exam at SSTC 2004 are eligible for a special discounted fee of $300.

For potential candidates in other parts of the world, the Computer Society also offers a CSDP training course, Software Engineering Overview, in our Distance Learning Campus. The course, available at www.computer.org/distancelearning/ for free to members, provides a comprehensive review of essential software engineering principles. ■

IEEE Recognizes NEC's Sekimoto with Medal of Honor

The IEEE has named former NEC Corp. Chair Tadahiro Sekimoto as recipient of the 2004 IEEE Medal of Honor. The award, sponsored by the IEEE Foundation, celebrates Sekimoto's "pioneering contributions to digital satellite communications, promotion of information technology R&D, and corporate leadership in computers and communications."

For more than 50 years, IEEE Life Fellow Sekimoto has led the digital communications research efforts of the Tokyo-based NEC Corp. Sekimoto designed early pulse-code modulation equipment, as well as coding and decoding circuitry. In the late 1960s, Sekimoto developed a time-division multiple-access system and an automatic routing system. Not only did his work have a considerable impact on satellite communications, but the technologies that Sekimoto developed later formed the foundation for modern cellular telephone networks.

Over the years, Sekimoto has been honored with many other awards, including the Grand Cordon of the Order of the Sacred Treasure from the Emperor of Japan, the IEEE Edwin Howard Armstrong Achievement Award, the Aerospace Communications Award from the American Institute of Aeronautics and Astronautics, and the IEEE Alexander Graham Bell Medal.

Nominations for the 2005 IEEE Medal of Honor are due by **1 July**. Nomination forms are available at www.ieee.org/about/awards/sums/mohsum.htm.

CALLS FOR IEEE CS PUBLICATIONS

Over the past couple of years, homeland security has become a major concern for governments worldwide. Government leaders have begun to focus on the need to protect both their citizenry and critical infrastructures including power systems, communications, government and military installations, and food and water supplies.

The Internet and related information technologies play a role in homeland security as both tools for defending infrastructures and as entities needing protection.

For a November/December 2004 special issue on homeland security, *IEEE Internet Computing* is soliciting original articles on the use of Internet and information technologies for homeland security and on the protection of critical technology assets. Suitable topics include risk assessment and recovery planning, the controlled sharing of sensitive information among organizations, sensor network-based early-warning systems, and surveillance, data aggregation, and data mining technologies.

Submissions are due **1 April**. The complete call for papers is available at www.computer.org/internet/call4ppr.htm.

OTHER CALLS

SRDS 2004, 23rd Symp. on Reliable Distributed Systems, 18-20 Oct., Florianópolis, Brazil. Submissions due **2 Apr.** www.srds2004.ufsc.br

ISSRE 2004, 19th Int'l Symp. on Software Reliability Eng., 2-5 Nov., Saint-Malo, France. Abstracts due **2 Apr.,** submissions due **18 Apr.** www.issre.org/2004/

ATS 2004, 13th Asian Test Symp., 15-17 Nov., Kenting, Taiwan. Papers due **15 Apr.** http://ats04.ee.nthu.edu.tw/~ats04/

LCN 2004, 29th IEEE Conf. on Local Computer Networks, 16-18 Nov., Tampa, Fla. Papers due **21 May.** www.ieeelcn.org

CALENDAR

APRIL 2004

4-7 Apr: ITSW 2004, 11th IEEE Int'l Test Synthesis Workshop, Santa Barbara, Calif. www.tttc-itsw.org

5-7 Apr: ITCC 2004, 5th Int'l Conf. on IT, Las Vegas, Nev. www.itcc.info

8-9 Apr: IWIA 2004, 2nd IEEE Int'l Information Assurance Workshop, Charlotte, N.C. www.iwia.org/2004/

14-16 Apr: COOL CHIPS VII, Int'l Symp. on Low-Power & High-Speed Chips, Yokohama, Japan. www.coolchips.org

14-16 Apr: ICECCS 2004, 9th IEEE Int'l Conf. on Eng. Complex Computer Systems, Florence, Italy. www.dsi.unifi.it/iceccs04/

15-17 Apr: IPCCC 2004, 23rd IEEE Int'l Performance, Computing, & Comm. Conf., Phoenix, Ariz. www.ipccc.org

18-21 Apr: DDECS 2004, 7th IEEE Workshop on Design & Diagnostics of Electronics Circuits & Systems Workshop, Tatranská Lomnica, Slovakia. www.ui.savba.sk/DDECS2004

19-22 Apr: CCGRID 2004, 4th IEEE/ACM Int'l Symp. on Cluster Computing & the Grid, Chicago. www-fp.mcs.anl.gov/ccgrid2004/

19-23 Apr: ASYNC 2004, 10th Int'l

Call for Articles for *Computer*

Computer seeks articles for a special issue on Internet data centers, to appear in November 2004. Guest editors are Krishna Kant from Intel and Prasant Mohapatra from the University of California, Davis.

Internet data centers form the backbone of most Internet-based services, including e-commerce, IP-based telecom services, hosting services, and the like. As the reach of the Internet widens and more business-critical services are offered, the demands on IDCs grow along multiple dimensions, including responsiveness, service differentiation, security, and availability. Many other forces are likely to effect how the data centers of the future are designed, provisioned, and operated. *Computer*'s special issue will focus on research issues in identifying and implementing new strategies for optimizing IDCs: application services, protocol enhancements, performance evaluations, provisions for adequate security, protection and isolation, and ensuring an adequate quality of service. *Computer* is soliciting a small number of high-quality papers from academia and industry that highlight various problems and solutions and provide a vision for future work in this area.

Topics of particular interest include system architecture and converged data centers; symmetric multiprocessors versus clustered systems; scalability, reliability, and fault-tolerance; performance evaluation and workload characterization; operations, control, and autonomic management; power management issues; exploitation of new hardware/software technologies; and issues of security, protection, and isolation.

The deadline for papers is **1 April**. Submission guidelines are available at www.computer.org/computer/author.htm. **Submit manuscripts at http://cs-ieee.manuscriptcentral.com/.**

Send inquiries to the guest editors at krishna.kant@intel.com and prasant@cs.ucdavis.edu.

Symp. on Asynchronous Circuits & Systems, Hersonissos, Crete. www.async04.gr

20-23 Apr: FCCM 2004, IEEE Symp. on Field-Programmable Custom Computing Machines, Napa, Calif. www.fccm.org

25 Apr: DBT 2004, IEEE Int'l Workshop on Current & Defect-Based Testing (with VTS 2004), Napa, Calif. www.cs.colostate.edu/~malaiya/dbt.html

25-27 Apr: EDP 2004, 10th IEEE/DATC Electronics Design Processes Workshop, Monterey, Calif. www.eda.org/edps/edp04/

25-29 Apr: VTS 2004, 22nd VLSI Test Symp., Napa, Calif. www.tttc-vts.org

26-30 Apr: IPDPS 2004, 18th Int'l Parallel & Distributed Processing Symp., Santa Fe, N.M. www.ipdps.org

MAY 2004

9-12 May: IEEE Symp. on Security and Privacy, Oakland, Calif. www.ieee-security.org/TC/SP-Index.html

10-13 May: ISEE 2004, Int'l Symp. on Electronics & the Environment, Scottsdale, Ariz. www.iseesummit.org

12-14 May: ISORC 2004, 7th IEEE Int'l Symp. on Object-Oriented Real-Time Distributed Computing, Vienna. www.vmars.tuwien.ac.at/isorc2004/

13-14 May: NATW 2004, IEEE 13th North Atlantic Test Workshop, Essex Junction, Vt. www.ee.duke.edu/NATW/

13-15 May: AQTR 2004, Int'l Conf. on Automation, Quality, & Testing Robotics, Cluj-Napoca, Romania. http://193.226.6.120/aqtr/

16-19 May: PADS 2004, 18th Workshop on Parallel & Distributed Simulation, Kufstein, Austria. www.pads-workshop.org/pads2004/

17-18 May: ICAC 2004, Int'l Conf. on Autonomic Computing (with WWW 2004), New York. www.autonomic-conference.org

19-21 May: VisSym 2004, Joint Eurographics/IEEE TCVG Symp. on Visualization, Konstanz, Germany. www.inf.uni-konstanz.de/cgip/VisSym04/

19-21 May: BIBE 2004, IEEE 4th Int'l Symp. on Bioinformatics & Bioengineering, Taichung, Taiwan, ROC. http://bibe2004.ece.uci.edu/

19-22 May: ISMVL 2004, 34th Int'l Symp. on Multiple-Valued Logic, Toronto. www.eecg.utoronto.ca/~ismvl2004/

23-28 May: ICSE 2004, 26th Int'l Conf. on Software Eng., Edinburgh, UK. http://conferences.iee.org.uk/icse2004/

24-27 May: ECBS 2004, 11th IEEE Int'l Conf. & Workshop on the Eng. of Computer-Based Systems, Brno, Czech Republic. www.fit.vutbr.cz/events/ECBS2004/

25-28 May: RTAS 2004, 10th IEEE Real-Time & Embedded Technology & Applications Symp., Toronto. www.cs.virginia.edu/rtas04/

26-27 May: SDD 2004, IEEE Int'l Workshop on Silicon Debug & Diagnosis, Ajaccio, France. Contact Mike Ricchetti, miker@intellitech.com.

JUNE 2004

2-4 June: PBG 2004, Symp. on Point-Based Graphics, Zurich, Switzerland. www.point-graphics.org

2-4 June: IWLS 2004, 13th Int'l Workshop on Logic & Synthesis, Temecula, Calif. www.iwls.org

5-9 June: SWTW 2004, Southwest Test Workshop, San Diego, Calif. www.swtest.org

7 June: CLADE 2004, Workshop on Challenges of Large Applications in Distributed Environments, Honolulu. www.caip.rutgers.edu/clade2004/

7-9 June: POLICY 2004, IEEE 5th Int'l Workshop on Policies for Distributed Systems & Networks, Yorktown Heights, N.Y. www.policy-workshop.org/2004/

12-15 June: WICSA 2004, 4th IEEE/IFIP Working Con. on Software Architecture, Oslo, Norway. http://wicsa4.cs.rug.nl

17-18 June: ICAC 2004, Int'l Conf. on Autonomic Computing (with WWW 2004), New York. www.autonomic-conference.org

19-23 June: ISCA 2004, 31st Ann. Int'l Symp. on Computer Architecture, Munich, Germany. http://wwwbode.cs.tum.edu/~isca/

21-24 June: CCC 2004, 19th Ann. IEEE Conf. on Computational Complexity, Amherst, Mass. www.cs.umass.edu/~barring/ccc2004/

Submission Instructions

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting. Complete instructions for submitting conference or call listings are available at www.computer.org/conferences/submission.htm.

A more complete listing of upcoming computer-related conferences is available at www.computer.org/conferences/.

23-25 June: IMSTW 2004, 10th IEEE Int'l Mixed Signals Test Workshop, Portland, Ore. www.ece.pdx.edu/imstw04/

23-25 June: MEMOCODE 2004, 2nd ACM/IEEE Conf. on Formal Methods & Programming Models for Codesign, San Diego, Calif. www.irisa.fr/manifestations/2004/MEMOCODE/

24-25 June: CBMS 2004, 17th IEEE Symp. on Computer-Based Medical Systems, Bethesda, Md. www.cvial.ttu.edu/Conferences/cbms2004/cbms2004.html

24-26 June: IWPC 2004, 12th Int'l Workshop on Program Comprehension, Bari, Italy. http://iwpc2004.di.uniba.it

27-30 June: ICME 2004, Int'l Conf. on Multimedia & Expo, Taipei. www.icme2004.org

27 June-2 July: CVPR 2004, IEEE Computer Soc. Conf. on Computer Vision & Pattern Recognition, Washington, D.C. http://cvl.umiacs.umd.edu/conferences/cvpr2004/

28 June-1 July: DSN 2004, Int'l Conf. on Dependable Systems & Networks, Florence, Italy. www.dsn.org

JULY 2004

6-9 July: ICWS 2004, IEEE Int'l Conf. on Web Services, San Diego, Calif. http://conferences.computer.org/icws/

6-9 July: CEC 2004, IEEE Conf. on E-Commerce, San Diego, Calif. http://tab.computer.org/tfec/cec04

7-9 July: ICPADS 2004, 10th Int'l Conf. on Parallel & Distributed Systems, Newport Beach, Calif. www.cacs.louisiana.edu/icpads2004/

19-23 July: ICPS 2004, ACS/IEEE Int'l Conf. on Pervasive Services, Beirut, Lebanon. http://icps2004.cse.ogi.edu/

AUGUST 2004

9-10 Aug: MTDT 2004, IEEE Int'l Workshop on Memory Technology, Design, & Testing, San Jose, Calif. Contact Rochit Rajsuman, r.rajsuman@advantest-ard.com.

19-20 Aug: ISESE 2004, Int'l Symp. on Experimental Software Eng., Redondo Beach, Calif. www.isese.org

30 Aug.-1 Sept: ICALT 2004, 4th IEEE Int'l Conf. on Advanced Learning Technologies, Joensuu, Finland. http://lttf.ieee.org/icalt2004/

SEPTEMBER 2004

6-10 Sept: RE 2004, 12th IEEE Int'l Requirements Eng. Conf., Kyoto, Japan. www.re04.org

11-17 Sept: ICSM 2004, 20th Int'l Conf. on Software Maintenance (with METRICS 2004, SCAM 2004, & WSE 2004), Chicago. www.cs.iit.edu/~icsm2004/

14-16 Sept: METRICS 2004, 10th Int'l Symp. on Software Metrics: The Science & Practice of Software Metrics, Chicago. www.swmetrics.org

15-18 Sept: SCC 2004, IEEE Int'l Conf. on Services Computing, Shanghai. http://conferences.computer.org/scc/2004/

20-23 Sept: CLUSTER 2004, IEEE Int'l Conf. on Cluster Computing, San Diego, Calif. http://grail.sdsc.edu/cluster2004/

20-24 Sept: WI-IAT 2004, IEEE/WIC/ACM Int'l Conf. on Web Intelligence & Intelligent Agent Technology, Beijing. www.maebashi-it.org/WI04/

20-25 Sept: ASE 2004, 19th IEEE Int'l Conf. on Automated Software Eng., Linz, Austria. www.ase-conference.org

27-29 Sept: VL/HCC 2004, IEEE Symp. on Visual Languages & Human-Centric Computing, Rome. http://vlhcc04.dsi.uniroma1.it/

OCTOBER 2004

5-8 Oct: ICNP 2004, 12th IEEE Int'l Conf. on Network Protocols, Berlin. www.icnp2004.de.vu

11-15 Oct: UML 2004, 7th Int'l Conf. on Unified Modeling Language, Lisbon, Portugal. www.umlconference.org

18-20 Oct: SRDS 2004, 23rd Symp. on Reliable Distributed Systems, Florianópolis, Brazil. www.srds2004.ufsc.br

25-27 Oct: MASS 2004, IEEE Int'l Conf. on Mobile Ad Hoc & Sensor Systems, Ft. Lauderdale, Fla. www.ececs.uc.edu/~cdmc/mass/

THE OHIO STATE UNIVERSITY, Columbus, Chair, Department of Electrical and Computer Engineering. The Department of Electrical and Computer Engineering at The Ohio State University invites applications and nominations for the position of Chair, which becomes available July 2004. Consistently ranked in the top 10 percent nationally, the Department of Electrical and Computer Engineering at The Ohio State University enjoys the tradition of an excellent reputation among electrical and computer engineering programs internationally. The department consists of 49 faculty members, including three members of the National Academy of Engineering, an Oldenberger Medal winner, and 16 Fellows of the Institute of Electrical and Electronics Engineers (IEEE). We also have several faculty members who are Fellows of other learned societies. Our faculty members have received the Presidential Early Career Award for Scientists and Engineers, the NASA Distinguished Public Service Medal, Presidential Young Investigator Awards, Office of Naval Research Young Investigator Award, and National Science Foundation CAREER Awards. OSU faculty have achieved international recognition for research in analog-digital integrated circuit design, communications, computer engineering, computer networks, computer vision, control, electromagnetics, electronic materials, high performance computing, optics, power engineering, robotics, signal processing, transportation, and wireless systems. Our strategic plan sets an ambitious but realistic agenda for the future with thrusts in advanced electronic materials and nanoscale devices, computational and distributed intelligence, and biotechnologies. Central to this plan are initiatives in multidisciplinary research; the department aggressively pursues these opportunities with, for example, the OSU Center for Materials Research, the College of Medicine, the Ohio Supercomputer Center, the Department of Biomedical Informatics and the Biomedical Engineering Center, and the Center for Automotive Research. It is expected that the next Chair will work to support and enhance interdisciplinary activities such as these. The successful applicant will present leadership skills and an extensive record of accomplishment in research and scholarship. The Ohio State University is committed to excellence in undergraduate and graduate education, and to diversity. Applications should include a curriculum vita, a statement of the candidate's vision for the future of ECE research and education, a self-assessment of leadership qualities and style, and the names of five professional references, all in the form of Word or pdf attachments (only) to the search committee chair, Prof. Kim L. Boyer, at: chairapp@ee.eng.ohio-state.edu. Review of applications will begin immediately and continue until the position is filled. The Ohio State University is an Equal Opportunity/Affirmative Action Employ-er; applications from women and other underrepresented groups are expressly encouraged.

THE UNIVERSITY OF TENNESSEE, The Imaging, Robotics, and Intelligent Systems (IRIS) Laboratory. The IRIS Lab invites applicants for multi-year Research Assistant/Associate Professorships and Ph.D. Fellowships. The IRIS Lab's emphasis is in the fields of Three-dimensional Imaging, Data Fusion, and Visualization. For 2004, the IRIS Lab is expected to have a staff of 50 and an annual budget over $3.5Million. Interested persons should contact: Mongi Abidi, Professor and Associate Department Head, Department of Electrical and Computer Engineering, 328 Ferris Hall, Knoxville, TN 37996-2100. Web: http://imaging.utk.edu/opportunities/opportunities.htm, Phone: 865-974-5454, Fax: 865-974-5459, E-Mail: abidi@utk.edu. UTK is an EE/AA/Title VI/Title IX/Section 504/ADA/ADEA Employer.

PRODUCT ENGINEER-MEMORY DEVICES: Develop and manage the entire life cycle of product development from product definition to market launch. BSc in Electronics Engineering or equiv req. Send ad w/resume to: Celetron USA Inc., 2125-B Madera Rd., Simi Valley, CA 93065.

NETWORK/SYSTEM ENGINEER in Birmingham, AL: Design, develop, improve & troubleshoot company's system and network, as well as related applications and databases. Manage & maintain load balance web servers. Setup & configure CISCO routers & firewalls for WAN/LAN. Req. MS or equiv. in Computer Sci./related field, 2 yrs exp.; Certification CCNA, MCSE, MCSD. Send resume to Billy Sanford, Intermark Group Inc. 1800 International Park Drive, Suite 500, Birmingham, AL 35243.

INTRATEL, LLC, located in Las Vegas, NV, seeks a full-time Software Engineer. The position requires a minimum Masters degree in Computer Science and 1 yr experience, including that in .NET framework, Voice over IP, digital signal process, and device driver programming. Competitive salary. Please send resumes via

The University of Memphis Teaching Fellowships

Teaching Fellowships will be awarded to 6 outstanding computer science graduate students for the 2004-2005 academic year. Fellows will assist local high school teachers in computer programming courses using innovative teaching techniques such as modeling software and Lego Robots. These fellowships are funded by National Science Foundation grant DGE-0338334 with stipends of $30,000 per year plus tuition. Fellowships are renewable up to two years. Awardees must be US citizens and hold a BS degree in computer science with experience in software engineering. Women and minorities are encouraged to apply. Applications should be received by April 2, 2004.

See www.cs.memphis.edu/tf for additional information and application.

SUBMISSION DETAILS: Rates are $275.00 per column inch ($300 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to: Marian Anderson, Classified Advertising, *Computer* Magazine, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; (714) 821-8380; fax (714) 821-4010. Email: manderson@computer.org.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *Computer* may reject any advertisement containing any of these phrases or similar ones: "…recent college grads…," "…1-4 years maximum experience…," "…up to 5 years experience," or "…10 years maximum experience." *Computer* reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. *Computer* assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. *Computer* encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

fax to Sivi Pederson, Dir. of Admin at 702-221-0904 or mail resumes to Intratel, LLC, 101 Convention Center Dr, Ste 700, Las Vegas, NV 89109, attn: Sivi Pederson.

ARCHITECTURE TECHNICAL SUPPORT ANALYST: BSc in Computer Science, related field or equivalent required. Send ad w/resume to: SunAmerica Inc., 1 SunAmerica Center, Los Angeles, CA 90067.

LOUISIANA STATE UNIVERSITY, Department of Computer Science, Assistant Professor, (Visualization and Computer Graphics). The Department of Computer Science, in collaboration with LSU CAPITAL (a new Center for Computation and Technology) invites applications for a faculty position with an emphasis in Visualization and Computer Graphics. The successful applicant will be expected to contribute to multidisciplinary projects with various application groups at LSU, with an emphasis on the use of grid or collaborative technologies. The position is expected to be filled in the Fall of 2004 at the Assistant Professor level, although more senior positions may be considered. LSU CAPITAL (www.capital.lsu.edu) is a new interdisciplinary research center with close ties to Computer Science; it receives approximately $9M per year in state funding. Much of this funding will be used to develop several new faculty positions with research groups in advanced technologies and computational sciences, spanning disciplines from engineering, basic and computer sciences, the arts, and business. The Center is actively pursuing advanced campus, state, and international networks, and an upgrade of its recently acquired 1024 processor Linux cluster. It is aggressively developing research groups in grids, visualization, software frameworks, collaborative environments, as well as many application disciplines, such as astrophysics, applied mathematics, biocomputation, nanotechnologies, and sensor networks. To enhance these activities, the Center has developed active visitor and fellowship programs. New faculty members will be expected to develop their own high-profile, interdisciplinary research programs that complement existing national and international CAPITAL projects, in addition to teaching duties. The Department of Computer Science at LSU offers the B.S., M.S. and Ph.D. degrees. The Department is the recipient of several new federal grants from the National Science Foundation for research in the area of Cyber Security and Sensor Networks. For details refer to the departmental web page at www.csc.lsu.edu. Candidates at the Assistant Professor level will be expected to show promise of significant future research activity and teaching effectiveness. The department values interdisciplinary research with impact and visibility beyond computer science. Successful candidates will be in a position to influence and contribute to the future directions of the department and CAPITAL. Required Qualifications: Ph.D. in Computer Science or a related field. Submit a letter of intent describing teaching and research interests along with a Curriculum Vitae (including e-mail address), and the names and addresses of three references to the address below. Search will continue until the position is filled. Electronic application is strongly preferred. Review of applications will begin March 15, 2004 and will continue until the position is filled. Chair, Faculty Search (faculty-vg) Committee, Depart-

THE UNIVERSITY OF TEXAS AT DALLAS

ERIK JONSSON SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

Faculty Positions

UTD

The Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas invites applications for senior faculty with an outstanding record of research, teaching and external funding in the areas of engineering and computer science that are considered interdisciplinary and/or emerging in nature and scope. The overall goal is to initiate new programs in areas including, but not limited to, biomedical engineering, bioinformatics, material science, chemical engineering and mechanical engineering. A startup package in seven figures has been budgeted to these positions.

The positions will be at the full professor level with tenure in the Erik Jonsson School of Engineering and Computer Science; starting spring, summer or fall 2004. Candidates must have a Ph.D. degree in Engineering, Computer Science or equivalent.

The Erik Jonsson School of Engineering and Computer Science currently offers B.S., M.S. and Ph.D. degrees in electrical engineering, computer science, software engineering and telecommunications engineering. The school also offers M.S. and Ph.D. degrees in computer engineering. There are 75 full-time tenure/tenure-track faculty in the Erik Jonsson School of Engineering and Computer Science. In fall 2002, a new 152,000 sq. ft. building opened for Computer Science and Engineering to supplement the existing 1994, 150,000 sq. ft. engineering and computer science building. The Engineering & Computer Science buildings provide extensive laboratory facilities for research in computer engineering, electrical engineering, telecommunications engineering, software engineering and computer science.

The University is located in the most attractive suburbs of the Dallas metropolitan area. There are over 900 high-tech companies within 5 miles of the campus, including Texas Instruments, Nortel Networks, Alcatel, Ericsson, Hewlett-Packard, Nokia, Fujitsu, MCI, EDS, and Perot Systems. Almost all the country's leading telecommunication's companies have major research and development facilities in our neighborhood. Opportunities for joint university-industry research projects are excellent. The Jonsson School has experienced very rapid growth in recent years and will become a top-ranked engineering school in the next five years. The Jonsson School is strengthening and expanding its programs by recruiting outstanding faculty and Ph.D. students, increasing funded research, and establishing new programs. The Jonsson School will benefit from a $300 million program of funding from public and private sources over the next five years (see www.utdallas.edu/utdgeneral/news/).

For more information, view the Internet webpage at **www.utdallas.edu/dept/eecs** or contact Dr. Duncan MacFarlane, Search Chair at 972-883-4658. The search committee will begin evaluating applications as soon as possible and will continue until the positions are filled.

Applicants should mail their resume with a list of at least five academic or professional references as soon as possible to:

Academic Search #754
The University of Texas at Dallas
P.O. Box 830688, M/S AD 23
Richardson, TX 75083-0688.

The University of Texas at Dallas is an Equal Opportunity Affirmative Action employer and strongly encourages applications from candidates who would enhance the diversity of the University's faculty and administration.

ment of Computer Science, 298 Coates Hall, Louisiana State University, Ref: Log #0693, Baton Rouge, Louisiana 70803. Ph: (225) 578-1495, Fax: (225) 578-1465. E-mail: search@csc.lsu.edu. LSU IS AN EQUAL OPPORTUNITY/EQUAL ACCESS EMPLOYER.

UNIVERSITY OF MEMPHIS. The Department of Electrical & Computer Engineering (www.ee.memphis.edu) at The University of Memphis is now accepting applications for one or more tenure track positions at the Assistant or Associate Professor levels. Review of applications will begin immediately and will continue until all positions are filled, with anticipated employment beginning August 2004. The University of Memphis is one of only two comprehensive public universities in the State of Tennessee and it is located in the largest urban center in the Mid-South. Specific departmental needs are in areas of computer engineering and electrical power systems; however, applicants from all areas of electrical and computer engineering will be considered. Successful applicants will be required to teach at both the undergraduate and graduate levels, and to initiate and sustain externally funded research. An earned doctorate in Electrical Engineering, Computer Engineering or a related area is required. Interested applicants should send resumes to: Faculty Search Committee, Department of Electrical & Computer Engineering, The University of Memphis, 206 Engineering Science Bldg., Memphis, TN 38152-3180. The University of Memphis is an EEO/AA employer. Under represented minorities are encouraged to apply. Successful candidates must meet Immigration Reform Act criteria.

TEXAS A&M UNIVERSITY-CORPUS CHRISTI, the Island University, located on the sunny South Texas coast, is seeking applicants for a tenure-track faculty position in computer science at the Associate/Professor level. TAMU-CC is the fastest growing university in Texas, and has been ranked the top public regional university in Texas two years in a row by U.S. News & World Report. The Department of Computing and Mathematical Sciences currently offers academic programs in computer science (BS, MS) that strongly emphasize the application of computing. We are planning to offer a Ph.D. in Applied Computing with an emphasis in environmental applications. We are seeking an individual with research interests in modeling, simulation, and visualization who has experience in advising Ph.D. students. The successful candidate will have the opportunity to play a major role in planning and developing our new Ph.D. program. A Ph.D. in computer science or closely related area is required. Applicants should send a letter of application, a curriculum vita, unofficial copies of graduate transcripts, and arrange to have three letters of recommendation sent to: College of Science & Technology, ATTN: Ken Brown, Texas A&M University-Corpus Christi, 6300 Ocean Drive (FC-179), Corpus Christi, TX 78412. The position is available Fall 2004. Primary consideration will be given to applications received by June 1, 2004. TAMU-CC is an Equal Opportunity Employer committed to diversity. http://www.tamucc.edu.

LOUISIANA STATE UNIVERSITY, Department of Computer Science, Assistant Professor, (Distributed Computing [Grid Computing] and Networks/one or more positions). The Department of Computer Science, in collaboration with LSU CAPITAL (a new Center for Computation and Technology) invites applications for a faculty position in Distributed/Grid Computing commencing Fall 2004. The successful applicant will be expected to develop a research program leading to the deployment of diverse applications on production computational grids. The position is expected to be filled at the Assistant Professor level, although more senior positions may be considered. LSU CAPITAL (www.capital.lsu.edu) is a new interdisciplinary research center with close ties to Computer Science; it receives approximately $9M per year in state funding. Much of this funding will be used to develop several new faculty positions with research groups in advanced technologies and computational sciences, spanning disciplines from engineering, basic and computer sciences, the arts, and business. The Center is actively pursuing advanced campus, state, and international networks, and an upgrade of its recently acquired 1024 processor Linux cluster. It is aggressively developing research groups in grids, visualization, software frameworks, collaborative environments, as well as many application disciplines, such as astrophysics, applied mathematics, bio-computation, nanotechnologies, and sensor networks. To enhance these activities, the Center has developed active visitor and fellowship programs. New faculty members will be expected to develop their own high-profile, interdisciplinary research programs that complement existing national and international CAPITAL projects, in addi-

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The Swiss Federal Institute of Technology Zurich is committed to expanding its impact in the area of Bioinformatics and invites applications for two faculty positions in the Departments of Biology and Computer Science. These positions will complement ongoing research and teaching activities at ETH, the Functional Genomics Center Zurich (FGCZ), and the ETH Laboratory of Computational Science and Engineering (CoLab). ETH Zurich and the University of Zurich will offer a joint Master's degree in Bioinformatics. The successful candidates will be expected to significantly contribute to this program.

Professor of Bioinformatics

Candidates are expected to have a strong, independent research program in applied bioinformatics. Research areas include the development of novel algorithms for the identification and analysis of functional modules, macromolecular interactions and molecular fluxes in living cells, using information available from genome sequences, three-dimensional protein structures, and data from proteomics and functional genomics experiments. The Department of Biology of ETH Zurich and Life Sciences Zurich offer outstanding scientific opportunities to participate in an interdisciplinary systems biology initiative and for collaborations with colleagues from other departments and from the University of Zurich.

Assistant Professor of Bioinformatics

Candidates should demonstrate exceptional potential to develop an innovative research program in bioinformatics. Specific research areas include genomics, molecular sequence analysis, molecular evolution, protein structure, gene expression, network analysis, computational proteomics, computational genetics, or structural and functional genomics and data visualization. The Department of Computer Science has a strong tradition in the area of computer systems and languages. The successful candidate will be expected to interface with the traditional research areas in the department and with the biological community. He or she will also be expected to participate in the Computer Science teaching program. The Assistant Professor position is non-tenure track and funded for six years.

Please submit your application together with a curriculum vitae, a list of publications, and a detailed research plan **to the President of ETH Zurich, Prof. Dr. O. Kübler, ETH Zentrum, CH-8092 Zurich, no later than March 31, 2004.** ETH Zurich specifically encourages female candidates to apply with a view towards increasing the proportion of female professors.

INFORMATICS iCORE CIRCLE OF RESEARCH EXCELLENCE

iCORE Research Chair
Distributed High Performance Computing

The Alberta Informatics Circle of Research Excellence (iCORE), in conjunction with Alberta universities, is seeking a Chair to lead a research program in the distributed systems, high performance computing, or grid computing areas. The Chair is expected to be an internationally recognized, exceptional researcher who will develop a research team in one or more of these areas.

Western Canada, through the WestGrid project (http://www.westgrid.ca), has invested $44 million in high performance computing research infrastructure that spans Calgary, Edmonton and Vancouver. These grid-enabled HPC resources are connected by a leading-edge dedicated optical network.

iCORE Chairs can be held at any of the three research universities in Alberta: University of Alberta, University of Calgary and the University of Lethbridge.

UNIVERSITY OF CALGARY · University of Lethbridge · UNIVERSITY OF ALBERTA

Successful applicants will be appointed with tenure and have substantial research funding for an initial period of five years, renewable once for a second five years.

Salaries and research funding associated with iCORE Chairs are highly competitive.

If you are interested, or know of someone who may be interested, please contact:

Lynn Sutherland, Vice President, Programs
(403) 210-5335 sutherland@icore .ca

www.icore.ca

Technical Director

Fraunhofer USA

Fraunhofer USA is searching for a Technical Director for its Center for Experimental Software Engineering (CESE) in College Park, Maryland. CESE is a fast growing, not-for-profit, applied research and technology transfer organization that has been in business since 1998.

Candidates for the position should hold the Ph.D. degree and have Industrial or academic research experience in software engineering with a commensurate publication record.

The Technical Director shares directorial responsibility with the Managing Director, provides technical leadership for the technical program of contracts and grants, and determines future research directions for the Center. The Technical Director is expected to be successful in obtaining contracts and grants, publish in appropriate conferences and journals, and be a presence in the software engineering research community.

The Center currently has contracts with federal agencies such as NASA and the DoD and with several local Maryland and international companies. Current activities involve software development process improvement, software capability evaluations, inspections, agile development, security and knowledge management. The Technical Director is expected to add to this list of capabilities.

Fraunhofer USA, an independent non-profit corporation, is a subsidiary of the Fraunhofer-Gesellschaft, Europe's largest organization for applied research. The goals of Fraunhofer USA are to facilitate cooperation between the Fraunhofer Institutes in Germany and high quality research programs in the U.S., and to provide customers with world class technology solutions.

Please send your resume to:
Frank Herman
Fraunhofer Center for Experimental Software Engineering
4321 Hartwick Rd., Suite 500
College Park, MD 20740
fherman@fc-md.umd.edu

D. E. Shaw Research and Development

Research on Algorithms and Architectures for Computational Biochemistry

Extraordinarily gifted computer scientists, systems architects, electrical engineers and systems software professionals are sought to join a rapidly growing New York–based research group pursuing an ambitious, long-term project aimed at achieving major scientific advances in the field of biochemistry and fundamentally transforming the process of drug discovery. This research effort is being financed by the D. E. Shaw group, an investment and technology development firm with approximately $6 billion in aggregate capital, and operates under the direct scientific leadership of its founder, Dr. David E. Shaw.

Among the group's current research activities is a project aimed at developing a massively parallel special-purpose supercomputer and innovative mathematical and computational techniques to direct unprecedented computational power toward the solution of key scientific and technical problems in the fields of molecular simulation and molecular design. Successful candidates will be working closely with a number of the world's leading computational chemists and biologists, and will have the opportunity not only to participate in an exciting entrepreneurial venture with considerable economic potential, but to make fundamental contributions within the fields of biology, chemistry and medicine.

Serious candidates will have an exceptionally distinguished history of academic and/or industrial accomplishment in computer science, electrical engineering, applied mathematics, or a related area. Particularly relevant areas of expertise might include parallel computation, high-speed interconnection networks, scientific computing, numerical analysis, optimization, the analysis of algorithms, operating systems, digital systems simulation, reconfigurable computing, and ASIC design, but specific knowledge of any of these areas is less critical than exceptional intellectual ability and a demonstrated track record of achievement. We are prepared to reward exceptionally well-qualified individuals with above-market compensation.

Please send your curriculum vitae (including list of publications, thesis topic, and advisor, if applicable) to Research.Development7@deshaw.com.

D. E. Shaw Research and Development, L.L.C. does not discriminate in employment matters on the basis of race, color, religion, gender, national origin, age, military service eligibility, veteran status, sexual orientation, marital status, disability, or any other protected class.

D E Shaw & Co

tion to teaching duties. The Department of Computer Science at LSU offers the B.S., M.S. and Ph.D. degrees. The Department is the recipient of several new federal grants from the National Science Foundation for research in the area of Cyber Security and Sensor Networks. For details refer to the departmental web page at www.csc.lsu.edu. Candidates at the Assistant Professor level will be expected to show promise of significant future research activity and teaching effectiveness. The department values interdisciplinary research with impact and visibility beyond computer science. Successful candidates will be in a position to influence and contribute to the future directions of the department and CAPITAL. Required Qualifications: Ph.D. in Computer Science or a related field. Submit a letter of intent describing teaching and research interests along with a Curriculum Vitae (including e-mail address), and the names and addresses of three references to the address below. Search will continue until the position is filled. Electronic application is strongly preferred. Review of applications will begin March 15, 2004 and will continue until the position is filled. Chair, Faculty Search (faculty-gc) Committee, Department of Computer Science, 298 Coates Hall, Louisiana State University, Ref: Log #0694, Baton Rouge, Louisiana 70803. Ph: (225) 578-1495, Fax: (225) 578-1465. Email: search@csc.lsu.edu. LSU IS AN EQUAL OPPORTUNITY/EQUAL ACCESS EMPLOYER.

UNIVERSITY OF BRIDGEPORT, Departments of Computer Science and Engineering, and Electrical Engineering Faculty Positions. The fast-growing departments of Computer Science and Engineering and Electrical Engineering at the University of Bridgeport invite applications for full time Instructor/Lecturer positions or tenure-track positions at the Assistant/Associate Professor levels. Candidates for tenure-track positions must have a Ph.D. in computer science, computer engineering, or electrical engineering. An M.S. degree is required for the Instructor/Lecturer positions. A strong interest in teaching undergraduate and graduate courses and an excellent research record are required. The ability to teach lab-based courses is also required. Applicants are sought in the areas of wireless design, programming languages, distributed computing, VLSI, communications, FPGA analysis, solid-state electronics, fiber optics, speech analysis, circuit theory, Image Processing, IC Design, Digital and Analog Controls, Medical Electronics, biomedical engineering and biometrics. There are opportunities to participate in the external engineering programs, which include weekend and evening graduate and continuing education classes, on-site instruction in local industry and distance learning initiatives. Applications and nominations will be accepted and considered until the position is filled with priority consideration given to those received on or before March 31, 2004. Applicants should send a cover letter, resume and address and e-mail address of four references to: Faculty Search Committee, School of Engineering, C/O Human Resources Department, Wahlstrom Library, 7th Floor, 126 Park Avenue, Bridgeport, CT 06601. Fax: (203) 576-4601, hr@bridgeport.edu. The University of Bridgeport is an Affirmative Action/Equal Opportunity Employer.

COMPUTER INFORMATION SYSTEMS MANAGER: Security, surveillance, and investigation company. 2 yrs exp req. Send ad w/resume to: Bradley G. Miller Investigators, Inc., 211 S. Beverly Dr. #108, Beverly Hills, CA 90212.

UNIVERSITY OF MIAMI. The Department of Electrical and Computer Engineering invites applications for a tenure-track faculty position in the area of wireless communication networks. The position is expected to be filled at the Assistant/Associate Professor level and will be available starting in the Fall semester of 2004. Qualifications include a Ph. D. degree in electrical engineering, computer engineering or computer science, evidence of scholarly accomplishment, the ability to initiate research projects, attract external funding and teach undergraduate and graduate courses. Relevant background and experience in multimedia systems and networks, including wireless networks, will be considered a definite advantage. Salary will be commensurate with rank and experience. The University of Miami is a private, independent and comprehensive research university with an enrollment of 13,400, of which 3,100 are graduate students and 1,900 are law or medical students. The College of Engineering is located in Coral Gables, an attractive suburb of Miami, Florida, and is in close proximity to a wide range of recreational and cultural activities. The Electrical and Computer Engineering Department is the largest department in the College of Engineering and offers the B. S., M. S. and Ph. D. degrees in both Electrical and Computer Engineering as well as a new program in Information Technology offering both undergraduate and graduate degrees. The ideal candidate would be expected to interface with existing research strength in multimedia systems and technology, wireless communications, telecommunications and information technology. Applications should be forwarded with a current curriculum vitae, an appropriate statement of career objectives and the names and addresses of at least three references to: Dr. James W. Modestino, Chair, Department of Electrical and Computer Engineering, University of Miami, P. O. Box 248294, Coral Gables, FL 33124-0640. The University of Miami is an equal opportunity/affirmative action employer.

UNIVERSITY AT BUFFALO. The Department of Computer Science and Engineering (CSE) at the University at Buffalo has an opening for a full-time non-tenure-track lecturer. Lecturers are expected to primarily teach and develop undergraduate CSE courses, and advise students. The minimum qualifications for the position are an MS degree in CSE or related field by Sept 2004, along with experience teaching courses for CSE majors as well as non-majors. Applicants should send by March 31, 2004 a cover letter, curriculum vitae, and the names of at least three references to: Lecturer Search Committee, CSE Department, 201 Bell Hall, Buffalo, NY 14260-2000. For more information about the position and the CSE Department, please visit http://www.cse.buffalo.edu. The University at Buffalo is an Equal Opportunity Employer/Recruiter.

LOUISIANA STATE UNIVERSITY, Department of Computer Science, Assistant Professor, (Security and Wireless Sensor Networks). The Department of Computer Science, in collaboration with LSU CAPITAL (a new Center for Computation and Technology) invites applications for a faculty position with an emphasis in Security, Reliability and Wireless Networking Infrastructure challenges posed by ubiquitous computing, sensor networks, a wide-area distributed computing and large scale web services. The position is expected to be filled in the Fall of 2004 at the Assistant Professor level, although more senior positions may be considered. This is one of several new faculty positions and represents a major strengthening of the department. LSU CAPITAL (www.capital.lsu.edu) is a new interdisciplinary research center with close ties to Computer Science; it receives approximately $9M per year in state funding. Much of this funding will be used to develop several new faculty positions with research groups in advanced technologies and computational sciences, spanning disciplines from engineering, basic and computer sciences, the arts, and business. The Center is actively pursuing advanced campus, state, and international networks, and an upgrade of its recently acquired 1024 processor Linux cluster. It is aggressively developing research groups in grids, visualization, software frameworks, collaborative environments, as well as many application disciplines, such as astrophysics, applied mathematics, bio-computation, nanotechnologies, and sensor networks. To enhance these

activities, the Center has developed active visitor and fellowship programs. New faculty members will be expected to develop their own high-profile, interdisciplinary research programs that complement existing national and international CAPITAL projects, in addition to teaching duties. The Department of Computer Science at LSU offers the B.S., M.S. and Ph.D. degrees. The Department is the recipient of several new federal grants from the National Science Foundation for research in the area of Cyber Security and Sensor Networks. For details refer to the departmental web page at www.csc.lsu.edu. Candidates at the Assistant Professor level will be expected to show promise of significant future research activity and teaching effectiveness. The department values interdisciplinary research with impact and visibility beyond computer science. Successful candidates will be in a position to influence and contribute to the future directions of the department and CAPITAL. Required Qualifications: Ph.D. in Computer Science or a related field. Submit a letter of intent describing teaching and research interests along with a Curriculum Vitae (including e-mail address), and the names and addresses of three references to the address below. Search will continue until the position is filled. Electronic application is strongly preferred. Review of applications will begin March 15, 2004, and will continue until the position is filled. Chair, Faculty Search (faculty-wsn) Committee, Department of Computer Science, 298 Coates Hall, Louisiana State University, Ref: Log #0692, Baton Rouge, Louisiana 70803. Ph: (225) 578-1495, Fax: (225) 578-1465. E-mail: search@csc.lsu.edu. LSU IS AN EQUAL OPPORTUNITY/EQUAL ACCESS EMPLOYER

INTRATEL, LLC, located in Las Vegas, NV, seeks a full-time Software Engineer. The position requires a minimum Masters degree in Computer Science and 2 years experience, including that in database administration and design, .NET programming, Web service, application and site implementation, business application implementation and multimedia service implementation. Competitive salary. Please send resumes via fax to Sivi Pederson, Dir. of Admin at 702-221-0904 or mail resumes to Intratel, LLC, 101 Convention Center Dr, Ste 700, Las Vegas, NV 89109, attn: Sivi Pederson.

THE UNIVERSITY OF TEXAS AT DALLAS

ERIK JONSSON SCHOOL OF ENGINEERING AND COMPUTER SCIENCE
COMPUTER ENGINEERING

Faculty Positions – System-Level Design, Computer/Processor Architecture, Real Time Systems and Electronic Design Automation

The Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas invites applications for computer engineering tenure/tenure-track faculty positions in:

- System-level design, synthesis and optimization; co-design, embedded systems
- Computer/processor architecture; high performance, power-aware superscalar and embedded processors; operating systems and compiler interactions with computer architecture; special purpose and DSP architectures
- Real time systems
- Electronic design automation; VLSI verification; synthesis

and related areas. Positions are at the assistant, associate or full professor levels, starting spring, summer or fall 2004. The successful candidate would be appointed a faculty position in either the Department of Electrical Engineering or the Department of Computer Science; a joint appointment is possible. Candidates must have a Ph.D. degree in Computer Science, Electrical Engineering, Software Engineering, Computer Engineering or equivalent. Candidates should have a strong record of research, teaching, and external funding. A significant start-up package has been budgeted for these positions.

The Erik Jonsson School of Engineering and Computer Science offers an interdisciplinary Ph.D. degree in Computer Engineering; M.S. Degree in Computer Engineering. Faculty for the computer engineering program consists of members from Computer Science and Electrical Engineering. Currently there are 18 CE affiliated faculty. In fall 2002, a new 152,000 sq. ft. building opened for Computer Science and Engineering to supplement the existing 1994, 150,000 sq. ft. engineering and computer science building. The engineering & computer science buildings provide extensive laboratory facilities for research in computer engineering, electrical engineering, telecommunications engineering, software engineering and computer science.

The University is located in the most attractive suburbs of the Dallas metropolitan area. There are over 900 high-tech companies within 5 miles of the campus, including Texas Instruments, Nortel Networks, Alcatel, Ericsson, Hewlett-Packard, Nokia, Fujitsu, MCI, EDS, and Perot Systems. Almost all the country's leading telecommunication's companies have major research and development facilities in our neighborhood. Opportunities for joint university-industry research projects are excellent. The Jonsson School has experienced very rapid growth in recent years and will become a top-ranked engineering school in the next five years. The Jonsson School is strengthening and expanding its programs by recruiting outstanding faculty and Ph.D. students, increasing funded research, and establishing new programs. The Jonsson School will benefit from a $300 million program of funding from public and private sources over the next five years (see **www.utdallas.edu/utdgeneral/news/**).

For more information, view the Internet webpage at www.ce.utdallas.edu or contact Dr. Duncan MacFarlane, Search Chair at 972-883-4658. The search committee will begin evaluating applications as soon as possible and will continue until the positions are filled.

Applicants should mail their resume with a list of at least five academic or professional references as soon as possible to:

Academic Search #755
The University of Texas at Dallas
P.O. Box 830688, M/S AD 23
Richardson, TX 75083-0688.

The University of Texas at Dallas is an Equal Opportunity Affirmative Action employer and strongly encourages applications from candidates who would enhance the diversity of the University's faculty and administration.

INFORMATICS iCORE CIRCLE OF RESEARCH EXCELLENCE

iCORE Research Chair
Distributed High Performance Computing

The Alberta Informatics Circle of Research Excellence (iCORE), in conjunction with Alberta universities, is seeking a Chair to lead a research program in the distributed systems, high performance computing, or grid computing areas. The Chair is expected to be an internationally recognized, exceptional researcher who will develop a research team in one or more of these areas.

Western Canada, through the WestGrid project (http://www.westgrid.ca), has invested $44 million in high performance computing research infrastructure that spans Calgary, Edmonton and Vancouver. These grid-enabled HPC resources are connected by a leading-edge dedicated optical network.

iCORE Chairs can be held at any of the three research universities in Alberta: University of Alberta, University of Calgary and the University of Lethbridge.

UNIVERSITY OF CALGARY · University of Lethbridge · UNIVERSITY OF ALBERTA

Successful applicants will be appointed with tenure and have substantial research funding for an initial period of five years, renewable once for a second five years.

Salaries and research funding associated with iCORE Chairs are highly competitive.

If you are interested, or know of someone who may be interested, please contact:

Lynn Sutherland, Vice President, Programs
(403) 210-5335 sutherland@icore .ca

www.icore.ca

Technical Director

Fraunhofer USA

Fraunhofer USA is searching for a Technical Director for its Center for Experimental Software Engineering (CESE) in College Park, Maryland. CESE is a fast growing, not-for-profit, applied research and technology transfer organization that has been in business since 1998.

Candidates for the position should hold the Ph.D. degree and have Industrial or academic research experience in software engineering with a commensurate publication record.

The Technical Director shares directorial responsibility with the Managing Director, provides technical leadership for the technical program of contracts and grants, and determines future research directions for the Center. The Technical Director is expected to be successful in obtaining contracts and grants, publish in appropriate conferences and journals, and be a presence in the software engineering research community.

The Center currently has contracts with federal agencies such as NASA and the DoD and with several local Maryland and international companies. Current activities involve software development process improvement, software capability evaluations, inspections, agile development, security and knowledge management. The Technical Director is expected to add to this list of capabilities.

Fraunhofer USA, an independent non-profit corporation, is a subsidiary of the Fraunhofer-Gesellschaft, Europe's largest organization for applied research. The goals of Fraunhofer USA are to facilitate cooperation between the Fraunhofer Institutes in Germany and high quality research programs in the U.S., and to provide customers with world class technology solutions.

Please send your resume to:
Frank Herman
Fraunhofer Center for Experimental Software Engineering
4321 Hartwick Rd., Suite 500
College Park, MD 20740
fherman@fc-md.umd.edu

D. E. Shaw Research and Development

Research on Algorithms and Architectures for Computational Biochemistry

Extraordinarily gifted computer scientists, systems architects, electrical engineers and systems software professionals are sought to join a rapidly growing New York–based research group pursuing an ambitious, long-term project aimed at achieving major scientific advances in the field of biochemistry and fundamentally transforming the process of drug discovery. This research effort is being financed by the D. E. Shaw group, an investment and technology development firm with approximately $6 billion in aggregate capital, and operates under the direct scientific leadership of its founder, Dr. David E. Shaw.

Among the group's current research activities is a project aimed at developing a massively parallel special-purpose supercomputer and innovative mathematical and computational techniques to direct unprecedented computational power toward the solution of key scientific and technical problems in the fields of molecular simulation and molecular design. Successful candidates will be working closely with a number of the world's leading computational chemists and biologists, and will have the opportunity not only to participate in an exciting entrepreneurial venture with considerable economic potential, but to make fundamental contributions within the fields of biology, chemistry and medicine.

Serious candidates will have an exceptionally distinguished history of academic and/or industrial accomplishment in computer science, electrical engineering, applied mathematics, or a related area. Particularly relevant areas of expertise might include parallel computation, high-speed interconnection networks, scientific computing, numerical analysis, optimization, the analysis of algorithms, operating systems, digital systems simulation, reconfigurable computing, and ASIC design, but specific knowledge of any of these areas is less critical than exceptional intellectual ability and a demonstrated track record of achievement. We are prepared to reward exceptionally well-qualified individuals with above-market compensation.

Please send your curriculum vitae (including list of publications, thesis topic, and advisor, if applicable) to Research.Development7@deshaw.com.

D. E. Shaw Research and Development, L.L.C. does not discriminate in employment matters on the basis of race, color, religion, gender, national origin, age, military service eligibility, veteran status, sexual orientation, marital status, disability, or any other protected class.

D E Shaw & Co

Apple Boosts XServe with PowerPC G5

Apple's Xserve G5, the third generation of its high-density 1U rackmount server, delivers more than 30 gigaflops of processing power per system—making it about 60 percent faster than its predecessor. It uses the same 64-bit PowerPC G5 processor that Virginia Tech uses in its cluster of Power Mac G5s, the world's third-fastest supercomputer.

Xserve G5 includes a new system controller with up to 8 Gbytes of PC3200 ECC memory; three hot-plug Serial ATA drive modules; optional internal hardware RAID; dual PCI-X slots, and dual on-board Gigabit Ethernet interfaces.

A cluster-optimized dual 2-GHz PowerPC G5 with 512 Mbytes of ECC RAM costs $2,999; www.apple.com.

Microsoft Upgrades MapPoint Web Service

MapPoint Web Service 3.5, the latest version of Microsoft's mapping technology for Web services integration, enhances capabilities for incorporating location data into a wide range of location-based systems. New capabilities include tools to help customers manage their corporate data and develop applications for mobile workers. MapPoint Web Service 3.5 also continues to offer support for Web standards such as SOAP and XML; www.microsoft.com/mappoint/webservice/.

InsiTech's Thin-Client Development Platform for Java

XML Tunneling Technology 4.0, released by InsiTech, is a rapid application development framework for building distributed Java applications. XTT supports thin-client development for applications used with relational databases and Web services. The framework is designed to extend Java development environments, such as Sun Java Studio and the NetBeans IDE.

XTT 4.0 costs $499; www.insitechgroup.com.

Please send new product announcements to products@computer.org.

New Version of Sybase's Mobile Development Tool

Pocket PowerBuilder 1.5 is an upgrade of Sybase's IDE for mobile and wireless enterprise applications. Listing at $1,295, Sybase Pocket PowerBuilder 1.5 is available through the end of March 2004 at a promotion price of $495; www.sybase.com.

PalmSource Announces New OS Developer Suite

The Palm OS Developer Suite, released by PalmSource, is designed to provide software developers an easier path toward creating Palm OS applications. The new suite includes an open-source IDE based on Eclipse. Originally developed by IBM, Eclipse offers several hundred plug-in tools that support many major languages, including C, C++, Java, and Cobol; www.palmsource.com.

StarOffice 7 Now Available for Solaris OS x86

Sun Microsystems has announced that its StarOffice 7 software, a multiplatform office productivity suite, is available for the Solaris OS x86 platform. The alternative desktop suite runs on the Solaris, Windows, Linux, and Windows operating systems and includes word processing, spreadsheet, presentation, drawing, and database capabilities.

StarOffice 7 for the Solaris OS x86 platform costs $79.195; www.sun.com.

Novell's XML-Based Visual Development Tool

Novell's exteNd 5 is a development suite to help companies integrate legacy systems and applications using Web services. New features in version 5 include visual tools to create interactive portals without writing code and support for industry standards such as the W3C XForms 1.0 specification.

Novell exteNd 5 components are available standalone, in a department/small business Professional Suite that costs $50,000 per CPU, or an Enterprise Suite that costs $120,000 per CPU; www.novell.com.

Expanded Interoperability for SecureZIP Technology

PKWare, inventor of the Zip compression format, has expanded the availability of its SecureZIP technology by licensing it to other vendors for the first time. The company has also released the PKZip Reader, a free utility that lets users open and decrypt SecureZIP files.

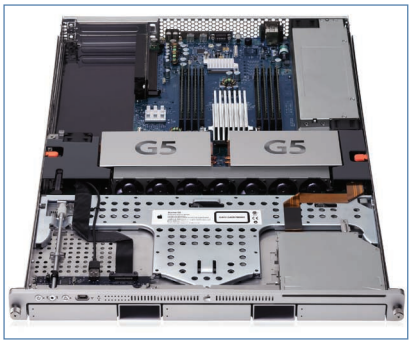
PKWare has already licensed the SecureZIP technology to its main competitor, WinZip, which will incorporate it into an upcoming beta release of its product to allow decryption of password-protected SecureZIP files.

The free PKZip Reader is available now for all Windows platforms; www.pkware.com/reader.

AppForge Ships Crossfire 5.0

AppForge has added a subscription program to Crossfire, the company's mobile and wireless application development environment for Microsoft's .NET platform. Under the new subscription program, customers receive continuous updates to ensure that their applications will run on the latest mobile devices.

Crossfire lets .NET developers develop a single application and deploy it to any number of Palm OS, Pocket PC, or Symbian devices. It includes built-in database support; www.appforge.com.



Available with either single or dual 2-GHz PowerPC G5 processors, the Xserve G5 architecture is based on an execution core that features massively parallel computation for an unprecedented 215 in-flight instructions.

Learning C++ and Java Together

Programming with Objects: A Comparative Presentation of Object-Oriented Programming with C++ and Java, Avinash C. Kak. The author compares and contrasts two of today's most popular programming languages, from basic constructs to their use in application-level programming for domains such as graphics, networks, and databases.

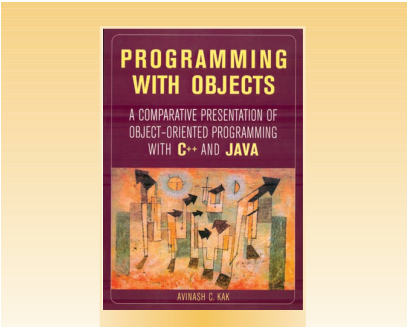
Given that both C++ and Java descend from C, learning these languages together offers several distinct advantages: It saves time and facilitates the mastery of each; learning by contrast and comparison can be more efficient and enjoyable; and writing a program in one language that corresponds to a given program in the other lets students tackle more difficult projects in either language.

The first half of the text covers basic language issues, while the second half details more advanced topics, including GUI programming, multithreading, and network and database programming.

Wiley-Interscience; www.wiley.com; 0-471-26852-6; 1,144 pp.; $79.95.

EMPOWERING DESIGN WITH RESEARCH

Design Research, Brenda Laurel. According to the author, designers can use design research tools to claim and direct the power of their profession. The new research models this book describes can help designers



investigate people, form, and process in ways that make their work more rewarding.

This book introduces the many research tools that can inform design and offers ideas about how and when to deploy them effectively. Chapter authors from locations including Stanford University, MIT, Intel, Maxis, Studio Anybody, and Sweden's HUMlab offer observations about how designers can make themselves better at what they do through research, illustrated with real-world examples that include case studies, anecdotes, and images.

MIT Press; mitpress.mit.edu; 0-262-12623-4; 336 pp.; $39.95.

TASK-ORIENTED TESTING

Best Practices for the Formal Software Testing Process: A Menu of Testing Tasks, Rodger D. Drabick. Software developers should not simply throw software over the wall to test engineers when coding is finished. A coordinated program of peer reviews and testing not only supplements a good software development process, it supports it.

This book presents a series of tasks to help organizations develop a formal testing process model, along with the inputs and outputs associated with each task. These tasks include review of program plans; development of the formal test plan; creation of test documentation; acquisition of automated testing tools; test execution; updating test documentation; and tailoring the model for projects of all sizes.

Dorset House Publishing; www.dorsethouse.com; 0-932633-58-7; 312 pp.; $41.95.

FROM VIRTUAL TO AUGMENTED REALITY

Virtual Applications: Applications with Virtual Inhabited 3D Worlds, Peter Andersen and Lars Qvortrup, editors. This collection of essays deals with the use of virtual inhabited 3D spaces in different societal domains. The trend now is to move from *virtual reality*—a reality into which users and objects from the real world should be moved—to *augmented reality*, in which computers move out into the world of real users, objects, and activities. The book also covers the use of virtual inhabited 3D spaces in both contexts.

The contributors examine VR and augmented reality use by analyzing the structure of application domains that use these technologies: production and manufacturing, communications support, scientific research, and artistic and cultural endeavors.

Springer; www.springer-ny.com; 1-85233-658-7; 272 pp.; $119.00.

Editor: Michael J. Lutz, Rochester Institute of Technology, Rochester, NY; mikelutz@mail.rit.edu. Send press releases and new books to *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; fax +1 714 821 4010; newbooks@computer.org.

Managing Systems Development

Gary Richardson and Blake Ives, University of Houston

realize that the process could be better executed; in fact, understanding of an optimal process did not exist. Management viewed system building as a "black art" and usually acknowledged that IT professionals were doing the best they could in difficult circumstances. Developers survived, not because they were efficient or effective, but because their products were important to the business and they were often the only source for IT-enabled solutions.

Missed deadlines, cost overruns, and failure to meet requirements are the rule rather than the exception for project development efforts. Two years ago, sportswear maker Nike made news by faulting supply-chain software deployment rather than the US economy for its earnings shortfalls.

Computerworld recently detailed 10 big-ticket project disasters totaling in the hundreds of millions of dollars in lost revenues. One firm, FoxMeyer Drugs, reportedly went into bankruptcy because of the failure of its enterprise resource planning system.

For both Nike and FoxMeyer, the culprits were third-party software packages. However, the risks escalate when design and custom coding enter the equation.

The Standish Group (www.standishgroup.com) has sponsored a project measurement survey for the past several years. For projects in 2001, average schedule overrun was 163 percent, average cost overrun was 145 percent, and actual functionality compared to plan was 67 percent. Only 26 percent of projects surveyed were judged a success; the lost value from marginal and failed projects was estimated at $75 billion.

This is a dismal report on the state of project management. The failures are rarely attributable to the underlying technology; recent research has identified a pattern of manageable factors that lead to both project success and failure. The bottom line is that effective management is usually the missing element in systems development.

Management must view IT project development as a business rather than a technical activity.

This article is the first of a two-part series on the unmet promise of IT project management. Here we describe the evolutionary path that has brought us to today's project culture and how successful organizations are managing projects. The second part of the series, to appear in the May IT Systems Perspective column, discusses four concrete strategies to improve project success.

EVOLUTION OF IT PROJECT MANAGEMENT

During the 1960s and the 1970s, systems development was a simple, informal process. Often, manually drawn flowcharts defined user requirements, and even less rigorous approaches were common. Programmers translated logic to code with little oversight. Project managers used haphazard methods to derive schedules and cost estimates. The results of these efforts—as measured in missed budgets, slipped schedules, and failed systems—reflected the lack of rigorous management.

Information technologists survived because enterprise management did not realize that the process could be better executed; in fact, understanding of an optimal process did not exist. Management viewed system building as a "black art" and usually acknowledged that IT professionals were doing the best they could in difficult circumstances. Developers survived, not because they were efficient or effective, but because their products were important to the business and they were often the only source for IT-enabled solutions.

In the 1980s, successful approaches began to emerge for structuring elements of the system development process. Tools and techniques evolved for eliciting requirements, estimating tasks, designing code and databases, and testing modules and systems.

Unfortunately, technological changes —particularly the migration of systems, first to minicomputers and then to desktops—dampened the impact of these advances. New architectural models coupled with changes in development tools and methodologies created a chaotic development environment. Managers became oversold on productivity tools, whose impact was largely unproven.

Nevertheless, management continued to view the resulting systems as strategic necessities. At the same time, it took notice of increasing development expenditures, cost overruns, and the associated business expenses of systems that failed to meet objectives.

Piecemeal approaches to project management continued into the 1990s. Some organizations differentiated themselves by consistently developing successful systems, but cost and schedule overruns and inadequate perfor-

mance remained the normal outcome for most firms. The underlying technology platforms, both for development and operating systems, continued to churn.

The first half of the decade was largely devoted to migrating to client-server architectures, while the Internet's phenomenal growth and Y2K conversions overshadowed project management. Some firms, burned by project disasters or losing confidence in their development units, turned to outsourcing as an easy, if sometimes shortsighted, solution.

KEYS TO PROJECT MANAGEMENT SUCCESS

A valuable lesson to emerge from these experiences is that, while new technologies can improve segments of the overall process, project success is linked to consistent task process and management involvement rather than to using a particular technology or canned methodology.

Far too often, system builders focus on preferred technical tools rather than on effectively managing the process. Careful predefinition, negotiation, and renegotiation of three essential project variables—functionality, cost, and schedule—are not high on the agenda of many project managers, many of whom are selected for their technical acumen rather than their project management skills.

Organizations with distinctive project-management competencies have largely achieved success by structuring their processes. For example, in his recent survey of the status of project management, C. William Ibbs, an expert in management processes for technical organizations, demonstrated that firms with more mature process models for system development showed dramatically improved accuracy in predicting project cost and schedule parameters. According to this model, within high maturity environments, average cost and schedule deviation was less than 8 percent of the original prediction as compared to the typical 200-percent variances for organizations whose development processes mirror those of previous eras.

Researchers at the Software Engineering Institute (www.sei.cmu.edu) have also observed varying levels of operational maturity in organizations. They likewise concluded that firms with higher levels of process maturity generally had lower costs of development.

Project success is linked to consistent task process and management involvement.

Despite the obvious need for and compelling benefits of funding and implementing standard processes, many organizations refuse to do so for two main reasons:

- Increasing the project maturity level requires more formalization and documentation and thus incurs greater front-end costs. Management is often unconvinced that these near-term investments will produce what they perceive as uncertain long-term benefits.
- Technically oriented managers continue to search for the silver-bullet technology or process change that will solve the development problem. Unfortunately, process changes can include what many incorrectly view as "extraneous" steps such as documentation, planning, status reporting, or postimplementation review.

If these problems are not resolved, senior management will become increasingly frustrated by its development organization's failures. Eventually this can, and often has, contributed to the decision to outsource development, thereby creating a new set of problems based on the IT investment's effectiveness.

Management must address the basic causes of project performance failures in a rational and prioritized manner. It must formulate projects to solve business problems and actively sustain the objective through development and implementation.

IT project development should be viewed as a business rather than a technical activity. Senior management must take ownership of this activity and even lead it in many cases. At the same time, IT staff must recognize that a project is not a technology proving ground; rather, they must work to predict economic as well as technical outcomes, meet user expectations, communicate status in a timely manner, and complete the activity as formally planned.

In the May column, we will describe a contemporary systems development model that will clarify the relevant work components and success characteristics that affect project outcomes. We will discuss the interrelationships of four processes—business alignment, system implementation, technical work, and project management—and describe how to begin the complex process of maturing project management organization. ■

Gary Richardson *is a visiting assistant professor and coordinator of the IT Project Management Certificate Program in the C.T. Bauer College of Business at the University of Houston. Contact him at gary.richardson@mail.uh.edu.*

Blake Ives *holds the C.T. Bauer Chair in Business Leadership in the C.T. Bauer School of Business at the University of Houston, where he is also director of the Information Systems Research Center. Contact him at blake.ives@uh.edu.*

Editor: Richard G. Mathieu, Dept. of Decision Sciences and MIS, St. Louis University, St. Louis, MO 63108; mathieur@slu.edu

Agent Hell: A Scenario of Worst Practices

Franco Zambonelli, University of Modena and Reggio Emilia
Michael Luck, University of Southampton

A little confusion goes a long way—too far—with software-based agents. Engineering discipline is the solution.

It was a pleasant, warm evening. Franco found a soft grassy area in the park, and sat down with a spaghetti ball and beer. He ate, drank, and rested quietly in the fading light of a beautiful spring evening. It wasn't a night to spend with the Salvation Army, he thought.

"Please stand up and report to the nearest police station! City laws forbid vagrants from sleeping in the park."

Startled by the harsh synthetic voice of the park control system, Franco looked up and saw one of those irritating all-look-the-same yuppies scrambling for a place to hide with his girlfriend. He saw, too, that—like every other boy of his age and social class that year—this one was wearing a pair of Timberland boots, carefully dirtied so he could appear as poor as possible.

Obviously, one of those ridiculously simple software agents that were dispersed in every corner of the city 20 years ago had never been deactivated. Now it had mistakenly identified the prosperous kid as a vagrant because of his scruffy boots.

AGENTS OF ENTHUSIASM

In 2017, a tidal wave of enthusiasm had launched a program to integrate automatic intelligent control systems. At first, the program had strong support from governments, suppliers, merchants, and consumers. By embedding agents in streets and other public sites, the city expected to provide useful information and improve public safety. In the park, for example, intelligent software agents controlled embedded sensors designed to recognize dangerous or illegal situations, discourage lawbreakers, and alert the nearest police station.

Franco didn't care much about the sensors. Like any true vagrant, he knew exactly how to outwit those software agents by simply rearranging his clothing. When the young boy gave up and left the park with his girlfriend, the synthetic voice stopped. Franco settled down again to watch the stars that began appearing in the sky.

Then, as often happened on such lonely nights, Franco started thinking of those last days of his youth—right before his personal agent hell began.

FRANCO'S DOWNFALL

On a Saturday night in July 2021, Franco had joined his friends at a café. He ordered a draft beer from a cheerful waitress, while his friend Paolo ordered one using a PDA. Everybody in the group began making bets on which of the two beers would come first.

Franco won the bet. Paolo paid the waitress for Franco's beer, ordered another one for himself, and then tried to track the status of his electronic beer order. The agent was stupidly smiling from the screen, asking Paolo to be patient, unable to say anything more.

Over the past few months, the waitress had consistently beaten the agent because of power reductions and congestion in e-marketplaces. In fact, most people no longer used a PDA to order a beer, but the fun of betting hadn't yet worn out with Franco and his friends.

Dynamic pricing among friends

At the same table, Andrea and Mario were talking about music. Mario was irritated because yesterday he had bought the latest Anastacia MP7 at almost twice the price Andrea paid just a few hours earlier. Such discrepancies were common with the auction-based pricing systems that e-marketplace service providers promoted as a technology destined to overtake human effort.

Paolo and Franco joined the discussion—yet another occasion to laugh about Paolo's PDA agent. The friends debated the recent European Commission resolution to impose strong price regulations on e-marketplaces. The resolution aimed to eliminate agent-based pricing systems.

Mario claimed to believe in a secret society that controlled the whole pricing system, but everyone knew this idea was just an urban myth. It was simply impossible to control the billions of agents negotiating in the network and the unbearable price fluctuations that resulted.

In the end, the friends all agreed on the usefulness of the EC's decision.

A missed connection

Laughing, talking, and drinking—a usual Saturday night. But Franco was expecting something unusual. At 10:00 p.m., he was to meet with Cecchetto, the city's most important music agent. Franco already had some success as a sax player in local discos and pubs. However, Cecchetto could change his life—opening doors to the big theaters, MP7 recordings for sale in major portals, and why not TV?

Franco had worked hard to arrange the meeting, which Cecchetto postponed several times. Now the time had come. But 10:00 passed, then 11:00, and Cecchetto did not appear. By midnight, Franco gave up.

He tried to contact Cecchetto the next day and the next, but couldn't get through to him. He began receiving messages from the discos and pubs where he usually played, saying, "We're sorry, but we have to cancel your performance." Soon, all his performances were cancelled, and he couldn't get any new gigs. Franco was without work or money. He eventually lost everything, including his old sax.

A mixed-up message

What happened was this: Cecchetto was simply running late that night and sent a personal message to Franco to apologize: "Franco, I am late. Please wait for me. I want your sax!"

Somehow, the last "a" of the message was misinterpreted as an "e." Franco had instructed his agents to automatically answer any sex-related spam messages, before deleting them, with messages like, "You moron! Stop wasting my time!"

Being very powerful and easily offended, Cecchetto shut every door in the city to Franco and his sax. Franco tried to explain what happened, but it was too late.

His story became one of many that led to the EC's decision a few years later to ban software agents.

AN INTERCONNECTED WORLD

Notable advances in both miniaturization and communications technology, together with advances in artificial intelligence and agent-based computing, enable us to imagine a world of pervasive computing technologies.

From heaven...

It is very likely that hardware technologies, properly empowered with agent-based software, will dramatically improve our quality of life. Applications in our homes and workplaces can use sensors and intelligent control systems to automatically perform tasks such as regulating room temperature and ordering supplies.

Global behaviors will become more important as agents begin to populate everyday environments.

At closer range, agents can coordinate the activities of wristwatches, PDAs, and cellular phones via short-range wireless technologies that interconnect devices worn on a person's body. Connecting such agents to a city's computer-based infrastructure could allow, for example, users of augmented reality glasses to visualize environmental dangers.

Most commercial transactions can occur in agent-based marketplaces, using computerized mechanisms and dynamic pricing systems to monitor trends that humans might not have the time or inclination to track.

...to hell

However, Franco's story points out the potential for agent technology to degenerate into agent hell.

For instance, the agents in the park scenario could not adapt effectively to changes in fashion, so a rich boy had to leave the park while Franco and other vagrants had learned how to cheat the system. Even if such agents could adapt to new situations over time, they would likely be useless—even disturbing—for long intervals.

Of course, deactivation is one way to circumvent a useless system. However, is it possible—or simply economically feasible—to remove millions of computer-based systems dispersed throughout a city? Alternatively, is it possible to deactivate the typically self-powered systems on which the agents reside? Or, to avoid deactivation, is it possible to globally reprogram millions of agents, forcing adaptation to a new situation?

The café scenario shows that agents do not necessarily improve the performance of even simple, useful tasks. Ordering a beer left Paolo's PDA agent stuck in the middle of a commercial network transaction.

Complexity and chaos. In complex and critical social mechanisms, such as pricing, agent-based systems could dramatically increase the instability and chaotic behavior that already characterize today's market economies.

In fact, some observers have claimed that the rigid rationality of an agent-mediated economy might provide more economic stability, but their claims are backed up by neither experience nor realistic simulations. Nor do they account for the unpredictable behaviors that can emerge in a collective. In the café scenario, the price differences in Mario and Andrea's music files may have emerged from the global agent-based economy having reached a strange—and possibly chaotic—attractor, regardless of any actual change in the demand for such goods.

In general, multitudes of interacting autonomous components executing in a dynamic environment suggest an interactive system in which the global state evolves as a function more of environmental dynamics and interactions than of internal component-level intelligence and rationality. Thus, as software agents begin to populate everyday networks and environments, global behaviors will become increasingly important in all agent-based activity.

Unfortunately, the state of the art in complexity science is still very far from offering constructive methods for con-

trolling global state in interactive systems. Without such methods, skeptics like Mario could easily reject agent systems as demonic entities under the control of an esoteric secret society.

Matters of trust. Delegating work to agents requires trusting them, yet software agent technology is unlikely to achieve the complex human decision-making capabilities that numerous tasks require. Franco's story is a possibly naive and extreme example of how the lack of these capabilities in a message agent could ruin someone's life.

Even with much more intelligent agents, trust is a difficult issue. While we do not argue that trusting agents is and will always be wrong, we do contend that trust must be achieved gradually. Potential advantages must be carefully evaluated against potential drawbacks.

Consumer and developer enthusiasm for advanced technologies already characterizes the software market. However, this enthusiasm can lead to shortcutting best practices in product development and test. Because agents are autonomous, deploying them with poor testing and documentation—in the tradition of some large software companies—could yield disastrous results. Instead, software agents should undergo exhaustive tests defining their characteristics and limitations, learning processes (if any), and behavior in relation to environmental dynamics and uncertainty—all carefully documented.

RESEARCH AGENDA

Our agent hell scenario aims to emphasize that it is not enough to explore methods of making agents more intelligent and autonomous or to analyze the ways and extent to which we can delegate work to them. Equally important is the need to advance the discipline related to engineering such systems. Agent-oriented software engineering research can address several areas that serve this end.

The research topics described here emerged explicitly from discussions at the meetings of the "Methodologies and Software Engineering for Agent Systems" Special Interest Group of the EC-funded Agentlink Research Network of Excellence (www.agentlink.org/activities/sigs/sig2.html).

Sociopolitical issues

We must study the social, political, and ecological implications of billions of agents executing in our physical environments, interacting with each other and with the environment in a globally interconnected network, and possibly capable of monitoring our everyday activities.

Prior to developing a software system, analysts should clearly understand its feasibility and likely impact. The pervasiveness and autonomous decision-making capabilities of agent-based systems make such considerations particularly important.

Environmental interrelationship

Environmental conditions change, but we may not necessarily have the option of updating an agent-based system's response to such changes.

We need to model the relationships of agent systems with their environments so that systems not only operate effectively and learn but also adapt to changing environmental conditions.

Tools and methodologies

To help develop and maintain well-engineered agent systems, we must define good modeling tools and methodologies. Despite the impossibility of controlling both individual agents and environmental dynamism, we need ways to predict interactive behavior among large numbers of agents and to provide some sort of control for easily maintaining them.

We believe that, because agent systems are large and closely tied to the physical world, good tools and methodologies should take their inspiration from the science of complexity and, more generally, from all scientific disciplines dealing with complex macro systems. This means adding physics, biology, and sociology to the logical sciences that traditionally dominate computer science.

Scalability and performance models

Finally, the need to study the scalability properties of multiagent systems, well before problems of scale arise, is both evident and pressing. Once agents begin to populate the world, their numbers will grow—resulting in potentially unmanageable systems of dramatic size. By then, it will be too late to rethink methodologies that we originally conceived for systems with only a few dozen agents.

We need new performance models specifically tuned to agent-based systems. Such models should do more than integrate and extend well-assessed performance models for distributed systems (which are, nevertheless, needed) and should also define performance models for trust to characterize how and to what extent we can trust an agent system to perform a given activity.

The moral of the story: Systematic development of agent-based systems requires rigorous software engineering processes and suitable tools to ensure a future agent heaven rather than agent hell. ■

Franco Zambonelli is professor in the Department of Engineering Science and Methodologies at the University of Modena and Reggio Emilia. Contact him at franco.zambonelli@unimore.it.

Michael Luck is professor in the School of Electronics and Computer Science at the University of Southampton. Contact him at mml@ecs.soton.ac.uk.

Editor: Sumi Helal, Computer and Information Science and Engineering Dept., University of Florida, P.O. Box 116125, Gainesville, FL 32611-6120; helal@cise.ufl.edu

Embedded Is the New Paradigm(s)

Wayne Wolf, Princeton University

A well-known set of applications has grown up around those capabilities: spreadsheets, word processing, and Web browsing—to name three. These are fundamentally important applications that won't go away any time soon.

The desktop computer won't disappear for the foreseeable future either. But we need to remember that as we assemble different components around the basic CPU and as we gang together powerful CPUs, the resulting system has new capabilities that we can use in new applications.

Embedded computing moved beyond toasters quite some time ago, but there are still misconceptions about what embedded computers do. Some of those misconceptions come from an old-fashioned view of what a computer is.

Embedded computing can support a world of applications not possible with mainframe or desktop interfaces.

MAINFRAME MONSTERS

The 1950s' science fiction movie *Forbidden Planet* is a prime example of one stereotypical view of computers. The movie introduced Robbie the Robot, but its main villain was a computer that was as big as the entire planet. This wasn't an embedded computer—it was the universe's biggest mainframe. The computer didn't come to the people; the people went to the computer.

We don't carry boxes of punch cards to the card reader any more, but viewing the computer as a distinct object still influences a lot of thinking about what computers are and what they can do.

Mainframes sat in a room and performed tasks like database management. The tasks involved large volumes of data, but the data was already in machine-readable form.

When powerful computers were expensive, it made sense to conserve their resources by paying people to prepare data for them.

DESKTOP DYNASTY

Desktop personal computers drastically lowered the cost of computing. Let's take a trip down memory lane.

Alto

The Alto set the form for today's desktop computer. Developed at Xerox Palo Alto Research Center in the 1970s, the Alto combined all the major components of today's PC: bit-mapped display, pointing device, local storage, and network connection.

The Alto's big insight was its attention to input and output. Most of Alto's CPU cycles went to I/O. The word processing program, for example, spent a lot of its time figuring out how to typeset the characters onto the screen and the printed page. Although the Alto didn't do things like handwriting recognition, this concept was very advanced and showed the way to a new vision of computing.

PC product category

However, it's important to keep in mind that "desktop PC" is fundamentally a product category. The computer itself is the CPU buried inside the box. The PC is a collection of components—the CPU, disk, screen, network connection, and so on—that combine to provide a specific set of capabilities.

From luggables to handhelds

Comparing two early portable computers emphasizes how system configurations and applications go hand-in-hand.

Osborne 1. Introduced in 1981, the Osborne 1 is generally credited as the first portable computer. It looked like a carry-on bag, and advertisers told us it was "so small it fits under an airline seat." (True, but what do you do with your feet?)

The Osborne was organized very much like a desktop PC. It had a 5-inch green CRT screen and two 5-1/4-inch floppy drives. It also had 120 V power cord—this machine didn't run on a battery.

You could run the same programs on your Osborne as you did on your desktop PC. That was its attraction but also its limitation. The Osborne had a relatively short lifespan. We had to wait a decade for desktop components to become sufficiently light and energy-efficient to make laptop computers popular.

TRS-80 Model 100. In 1983, Radio Shack introduced the TRS-80 Model 100, a portable computer designed for one application: text capture.

It had an LCD screen, a built-in keyboard, and a port for attaching a modem. The CPU was the Intel 80C85, a CMOS processor. CMOS was relatively rare at the time but consumed considerably less power than the standard NMOS processor. It ran on four AA batteries and came with a built-in word processing program, address book, scheduler, and Basic.

The Model 100 was wildly popular with reporters, and for several years it was among the world's best-selling computers. It was popular because it was tailored to an application and performed that job well. Reporters needed to be able to write text, edit it, and then deliver it to their editors while on the road.

The Model 100 didn't have much RAM or any disk drive. You couldn't store a book on it, but you could write an article like this one with the entire computer sitting on your lap.

I have both these machines in my personal collection of "antique" computers. By today's standards both look old-fashioned—one of my students reacted to the Osborne by saying, "You mean that's from the 80s?"

But the Model 100's small form factor and clean design come much closer to a modern device. I can imagine using the Model 100 because it fits my needs. I can't imagine holding the 25-pound Osborne on my lap.

Personal Digital Assistants. The PDA is a modern version of the Model 100. It's PC-like in some ways—Windows CE devices run spreadsheets and word processors, for example—but it is designed as a mobile device for tasks you want to do while on the move.

A PDA performs real-time handwriting recognition. It is also highly optimized to reduce energy consumption and stretch battery life.

Moreover, PDAs are designed to work in tandem with PCs. This design feature distinguished the original Palm device from previous handheld computers. You can manipulate your schedule and phone book either on the desktop or on the PDA, and you can move information seamlessly back and forth between them.

The PDA is optimized for mobile use and the more general, feature-rich user interface is reserved for the desktop.

Gesture control is a prime example of computation serving the user rather than vice versa.

CPU CAPABILITIES

But CPUs can do more than manipulate preformatted data or expand desktop capabilities. We need to go beyond the desktop to wherever the action is, whether on the road, in the sky, or on the water.

So what can we use CPUs for?

We can certainly employ them in user interface functions, such as handwriting recognition in a PDA.

We can also use them to process all sorts of streaming data. Data came into mainframes in batches, but the real world operates continuously. Today's powerful embedded CPUs can handle this constant data barrage if we're clever enough to exploit their capabilities.

We can, of course, use embedded computers to do things behind the scenes. CD and DVD players are prime examples of a trend toward using computation to correct for mechanical device limitations.

CD and DVD drives are cheap, flimsy plastic devices. Reading the data off the disk requires controlling that little piece of plastic to within micron tolerances. CD and DVD players do this by using embedded processors to execute complex control algorithms that operate continuously on the data streaming in from the read head. The typical CD player performs ridiculous amounts of computations using the power available from a couple of batteries.

Communications is another enabling technology for advanced systems. Once again, we increasingly use computation to overcome the limitations of the transmission medium. Embedded computing makes it possible to deploy sophisticated communication algorithms on battery-powered devices like cell phones.

More complex communication systems, such as those in which there are multiple antennas, will rely even more on embedded processing.

WHAT DO USERS WANT?

But let's get back to what the user wants to do with all these widgets.

As this column's title suggests, there is no single paradigm for embedded computing. Many applications use it, and each one imposes different requirements on the embedded system. Embedded computing's explosive growth only means that it has matured enough to support applications that weren't possible with mainframe or desktop interfaces.

Gesture recognition

Gesture-controlled interfaces represent one radical departure from the desktop paradigm. Our research group at Princeton has experimented with devices based on this technology, and I saw an IBM demonstration of gesture-based control a few months ago. The MIT Media Lab and many others have developed gesture-based control systems, and Sony recently announced an application for the PlayStation 2.

Gesture control is a prime example of computation serving the user rather than vice versa. Performing even modest gesture-recognition algorithms takes a lot of compute power, but abundant, low-cost embedded processors can support complicated devices to make the user's life simpler.

Inventory control

In a more prosaic vein, the communication systems that shipping and trucking companies use to track inventory provide another good example of

computation in the service of users. The underlying function is a fairly straightforward database. The problem is getting the data into and out of that database. So drivers carry battery-powered handhelds and use terminals in their trucks to track deliveries automatically.

Radio-frequency identification tags will take these systems to new levels. For example, RFID tags allow automatic readers to track inventories without requiring people to punch keys or swipe bar codes.

Scientific computing

Scientists and engineers shouldn't be chained to their computers any more than business people, nor should they be restricted to batch-mode analysis. Embedded computing can allow data processing, analysis, and visualization to track with scientific data collection.

The real-time and low-power challenges of next-generation scientific computing will require solutions that apply embedded system techniques.

Distributed processing

All these applications involve important distributed computing functionality. We generally don't solve embedded computing problems with one big CPU. Instead, we use a network of distributed processors to put the computation where it is needed.

There are several reasons to use distributed over centralized approaches—the two most potent are real-time deadlines and energy consumption.

Distributed computing complicates system design, but many applications require it. While all the traditional distributed computing methods are useful in embedded systems, we need to extend them to handle real time, limited bandwidth, and energy constraints.

Some people regard almost any information system as a desktop-style computer attached to some magic device. That view not only restricts our notion of a user interface but also ignores the design problems that all the other parts of the system pose.

The boundary between user interface and computation is increasingly blurred. Interpreting gestures, for example, requires a fairly abstract model of a person and the system state. Interpreting user needs requires more than just pixel pushing. Distributed architectures that use embedded software technology will perform all sorts of functions to meet this need. ■

Wayne Wolf *is a professor of electrical engineering at Princeton University. Contact him at wolf@princeton.edu.*

Languages and the Computing Profession

Neville Holmes, University of Tasmania

Around Christmas, feeling the need for some light technical reading and having long been interested in languages, I turned to a story in *Computer*'s Technology News department (Steven J. Vaughn-Nichols, "Statistical Language Approach Translates into Success," Nov. 2003, pp. 14-16). Toward the end of the story, the following paragraph startled me:

> Nonetheless, the grammatical systems of some languages are difficult to analyze statistically. For example, Chinese uses pictographs, and thus is harder to analyze than languages with grammatical signifiers such as spaces between words.

First, the Chinese writing system uses relatively few pictographs, and those few are highly abstracted. The Chinese writing system uses *logographs*—conventional representations of words or morphemes. Characters of the most common kind have two parts, one suggesting the general area of meaning, the other pronunciation.

Second, most Chinese characters are words in themselves, so the space between two characters is a space between words. True, many words in modern Chinese need two and sometimes more characters, but these are compounds, much like English words such as *password, output,* and *software.*

Translating natural language is too important and complex for computing professionals to tackle alone.

Third, Chinese does have grammatical signifiers. Pointing a browser equipped to show Chinese characters at a URL such as www.ausdaily.net.au will immediately show a wealth of what are plainly punctuation marks.

Fourth, Chinese is an isolating language with invariant words. This should make it very easy to analyze statistically. English is full of prefixes and suffixes—the word *prefixes* itself has one of each—which leads to more difficult statistics.

I do not mean these observations to disparage the journalist who wrote the story—but they do suggest that some computing professionals may know less than they should about language.

LANGUAGE ANALYSIS

The news story contrasted two approaches to machine translation.

> Knowledge–based systems rely on programmers to enter various languages' vocabulary and syntax information into databases. The programmers then write lists of rules that describe the possible relationships between a language's parts of speech.
>
> Rather than using the knowledge-based system's direct word-by-word translation techniques, statistical approaches translate documents by statistically analyzing entire phrases and, over time, 'learning' how various languages work.

The superficial difference seems to be that one technique translates word by word, the other phrase by phrase. But what one language deems to be words another deems to be phrases—agglutinative languages mildly so and synthetic languages drastically so—compared to relatively uninflected languages like English. Also, the components of a phrase can be contiguous in one language and dispersed in another—as in the case of German versus English as Samuel Langhorne Clemens described (www.bdsnett.no/klaus/twain).

The underlying difference seems to be that the knowledge–based systems' data for each language comes from grammarians, while the statistical systems' data comes from a mechanical comparison of corresponding documents, the one a professional translation of the other.

LANGUAGE TRANSLATION

Looking at translation generally, the problem with the statistical approach is that it requires two translation programs for every pair of languages: one

Continued on page 102

Continued from page 104

going each way. Ab initio, the same is true of the grammatical approach.

The number of different languages is such that complete coverage requires numerous programs—101 languages would require 10,100 translation programs. Daunting when we consider the thousands of different languages still in popular use.

The knowledge–based or grammatical approach provides a way around this. If all translations use a single intermediate language, adding an extra language to the repertoire would require only two extra translation programs.

The news story does describe a similar approach, a *transfer system*, but this uses a lingua franca as the intermediate language, which in part is probably why it has been found unsatisfactory. The other unsatisfactory aspect is commercial—the extra stage when the commercial enterprise seeks merely to translate between two written languages adds extra complexity and execution time.

To cope with the variety of and within natural languages, a completely unnatural language must serve as the intermediary. Designing this intermediate language would be a huge and difficult task, but it would reap equally huge benefits.

Without this approach to machine translation, it would be difficult and expensive to cater for minor languages, to make incremental improvements as individual languages change or become better understood, and to add parameters that allow selection of styles, periods, regionalities, and other variations. When the translation adds conversion between speech and text at either end, adopting the intermediary approach will become more important, if not essential.

INTERMEDIATE LANGUAGE

The intermediate language must be like a semipermeable membrane that lets the meaning pass through freely while blocking idiosyncrasies. Although designing and managing the intermediary would be a nearly overwhelming task, certain necessary characteristics suggest a starting point.

- *Specificity.* Every primary meaning must have only one code, and every primary code must have only one meaning. The difficulty here is deciding which meanings are primary.

Global acceptance of an auxiliary language would foster the disappearance of minor languages.

- *Precision.* A rich range of qualifying codes must derive secondary meanings from primary meanings and assign roles to meanings within their context.
- *Regularity.* The rules for combining and ordering codes, and for systematic codes such as those for colors, must be free from exceptions and variations.
- *Literality.* The intermediate language must exclude idioms, clichés, hackneyed phrases, puns, and the like, although punctuational codes could be used to mark their presence.
- *Neutrality.* Proper names, most technical terms, monocultural words, explicit words such as *inkjet* when used as shown here, and possibly many other classes of words must pass through the intermediate language without change other than, when needed, transliteration.

My use of the term "code" in these suggested characteristics, rather than *morpheme* or *word*, is deliberate. Designing the intermediate language to be spoken as words and thus to serve as an auxiliary language would be a mistake.

First, designing the intermediate language for general auxiliary use would unnecessarily and possibly severely impair its function as an intermediary. Second, a global auxiliary language's desirable properties differ markedly from those needed for an intermediary in translation, as the auxiliary language Esperanto's failure in the intermediary role demonstrates.

Indeed, given the possibility of general machine translation, it is possible to make an argument against the very idea of a global auxiliary language. Natural languages—the essence of individual cultures—are disappearing much faster than they are appearing. Global acceptance of an auxiliary language would foster such disappearances. Versatile machine translation, particularly when speech-to-speech translation becomes practical, would lessen the threat to minor languages.

WORK TO BE DONE

Defining the intermediate language requires developing and verifying its vocabulary and grammar as suitable for mediating translation between all classes and kinds of natural language.

The *vocabulary*—the semantic structure, specifically the semes and their relationships—will in effect provide a universal semantic taxonomy. The semes would be of many different kinds, both abstract and concrete. A major challenge will be deciding which meanings are distinct and universal enough to warrant their own seme and where to place them in the seme hierarchy. The key professionals doing such work will be philosophers and semanticists.

The rules for associating and separating semes and seme clusters, the *grammar*, would encompass the work of punctuation, although much of the meaning found in natural-language punctuation could be coded in the intermediate language's semes, unless implied by the language's grammar. The intermediary grammar might need to designate some semes—for example, some of the two dozen or so meanings given for the term "the" in the *Oxford English Dictionary*—as required to be inferred if they are not present in the source language. The key professionals in this work will be translators, interpreters, and linguists.

When involved in a project to develop an intermediary language, these two groups of professionals will need to work closely together, as grammar and vocabulary are closely interdependent. In this case, both must cope with the translation of many hundreds of wildly different languages.

What role would computing professionals play in such a team? Given the project's purpose—to make general machine translation possible—computing professionals would be of vital importance, but in a supporting role. Using different approaches to evaluate the intermediate language and its use for a variety of languages would require a succession of translation programs.

Those involved in this project will need to consider how to keep Web pages in both their original language and the intermediary so that browsers could, if necessary, translate the page easily into any user's preferred language. Allied to this requirement would be consideration of how to index the intermediary text so that all of the Web's content would be available to searchers. Indeed, the qualities of an intermediate language could make search engines much more effective.

Translation of SMS messages and e-mail should also be studied; ultimately, use of the intermediate language in telephones for speech translation should become possible. Users would select the natural language to use on their phone. The translation might then be through text, staged with speech-to-text conversion, or the processor might convert speech directly to or from the intermediate language. In any case, intermediary codes would be transmitted between users' phones and thus the language of one user would be independent of another user's.

General use of such speech translation would trail text translation by a long way, but even general text translation would promote global cooperation, providing an excellent return on investment in the project.

I began this essay when reports from the UN Forum on the Digital Divide in Geneva first became public. The failure of this beanfeast was both predictable ("The Digital Divide, the UN, and the Computing Profession," *Computer*, Dec. 2003, pp. 144, 142-143), and a scandalous waste of money given the number of poor in the world dying daily of hunger or cheaply curable illnesses.

Strategically, a much better way to use digital technology to help the poor and counter global inequity and its symptomatic digital divide would be for the UN to take responsibility for the development and use of a global intermediate translation language. International support would be essential, both to make swift development possible and, more importantly, to protect the work from intellectual-property predators.

Success would make truly global use of the Internet possible. Ultimately, with translation and speech-to-text conversion built into telephones, UN and other aid workers could talk to the economically disadvantaged without human interpreters.

However, an intermediate language project such as this could not be contemplated without the strong and active support of various professional bodies, particularly those from the fields of computing, philosophy, and language. Computing professionals should work with others to get public attention for this project and ensure that the needed professional support is made available. ■

Neville Holmes *is an honorary research associate at the University of Tasmania's School of Computing. Contact him at neville.holmes@utas.edu.au. Details of citations in this essay, and links to further material, are at www.comp.utas.edu.au/users/nholmes/prfsn.*