Core Technologies in Hardware and Software

Bruce Shriver

Genesis 2, Inc.

University of Tromso, Norway

Hong Kong University

Overview

- A definition, a question, and a thesis
- Commodities
- Constraints
- New market segments
- Core technologies: back to the future
- Closing observations

Core technologies

- Definition: A core technology is a technology that is both central and critical to what it is that an organization does.
- Central question: What are the core technologies for the products and services that you will provide in 3-5 years?
- Thesis: The way we design, implement, test, deploy, maintain, and evolve products and services is subject to relentless and significant change.

Some implications

- How and where we add value to systems is undergoing dramatic change
- How and where vendors differentiate themselves from one another is undergoing dramatic change

The myth of legacy

- There are entirely new markets being created for microprocessor-based systems where vendors do not have to support either legacy operating systems or legacy applications

Platforms as commodities

The design and implementation context (technology pull and application push)

The convergence of computing, communications, and entertainment, process technology and Moore's Law, utilizing 100-Million transistors on a chip, making the computer industry a commodity-based industry, distributed and parallel systems, the co-existence of client/server and peer/peer systems, distributed and parallel systems, embedded systems, real-time operating systems, GPS, xDSL, multi-GByte per second network interfaces, privacy and security, power, process and packaging technology, memory and cache architecture and technology, compiler technology, knowledge of the execution behavior of complex systems, design, thin clients, "smart" smart cards, simulation, and synthesis technology tools, ILP theory and practice, etc.

Commodity platforms

- Desktop boxes
- Workstations
- N-way SMP servers
- Laptops
- Clusters
- High-performance peripherals
  - High-capacity disk drives, printers, scanners, network interfaces, communications devices, CD-ROMs, DVDs, etc.

The driver: economies of scale

- From modest volumes with high margins
- To enormous volumes with thin margins
- We've moved from commodity hardware and commodity software to *commodity systems*
  - LANs-in-a-box
  - SMP-servers-in-a-box
  - Clusters-in-a-box

Application driven demands

- Relentless demand for high-performance processors
  - Superscalar, pipelined microarchitectures
  - Embedded and special purpose microarchitectures
- Relentless demand for high-performance servers and glueless SMP
  - From commodity 4-way SMP we are en route to commodity 32-way
  - Cache-coherency protocols and precise exception handling protocols

Microprocessor design and implementation constraints

- Power, process and packaging technology
- Bus, memory and cache architecture and technology
- Compiler technology
- Knowledge of the execution behavior of complex systems
- Design, simulation, and synthesis technology tools
- ILP theory and practice

Additional constraints

- Pinouts, core logic, motherboard design and implementation requirements, BIOS requirements, heat dissipation, etc.
- Industry "standards"
- Intellectual property related issues (patents, copyrights, and trade secrets)

MICRO-XY, ISCA, HPCA Topics

Examples

Dealing with some of the design and implementation constraints

Dealing with the power hegemony
(from notes taken at Fred Pollock's keynote on Tuesday)

- Current approaches
  - Low-power circuit and μArch techniques
  - SIMD ISA instructions
  - On-die L2-caches
  - Multiple CPU cores on a die
  - Multithreaded core and on-die L2-cache
- Future directions
  - Special purpose performance
  - Increased execution efficiency
  - Breaking the dataflow barrier

ILP and μarch: theory and practice (an incomplete list of issues)

- Fetching, decoding, issuing, scheduling, executing, completing, and retiring multiple operations in parallel
- In-order, out-of-order, and combinations of in-order and out-of-order of all of the above execution phases with the constraint of in-order retirement
- Branch and value prediction, speculative execution

Incomplete list … (cont.)

- Dealing with hazards (data, procedural, resource conflicts, and others); renaming state; exception and error handling
- Different design philosophies: e.g., various degrees of static and dynamic binding
- Multithreading and related issues
- Etc.

Dealing with a processor vs. platform/systems centric view

- What memory and cache architectures are required to minimize the latency and to sustain the required bandwidth for the target performance goal?
- What bus architectures are required? What speed buses are needed for the target market functionality? How many should there be? Are special high bandwidth ports required?

Processor vs. platform/systems centric view (cont.)

- What processor and core logic interface functionality must exist? At what clock speeds and bandwidth? Must they support pipelined access?
- What functionality should be integrated onto the motherboard? Onto the chip? E.g., see Intel's Timna and other on-chip integration and related system on a chip efforts.
- What are the implications to the answers to these (and similar) questions w.r.t. to µArch?

µArch Example 1: processing traces

- Encapsulating dynamic instruction sequences into "traces" at run time
- Traces become the fundamental "unit" of work
  - Traces typically contain multiple basic blocks
  - Often contain 16 instructions
- Traces are fetched, cached, and executed as a unit

Processing traces (cont.)

- High instruction fetch rates
- Needs very wide-issue execution engines
- Multiple processing elements are employed
  - Each PE = superscalar execution unit

Trace-level optimizations

Jacobson & Smith

Rotenberg & Smith

Friendly, Patel, and Patt

µArch Example 2: simultaneous multithreading

- Wallace, Tullsen, and Calder
- Parcerisa and González
- Tullsen, Lo, Eggers, and Levy
- Hily and Seznec

Pre-silicon development environments

- Enable IHVs and ISVs to develop products that will be tested and available when the microprocessor ships in production volumes
- Provide “cross-system” development tools if the processor has a new ISA (e.g., IA-64)
- Are absolutely essential and should be
  - Based on non-proprietary, open system interfaces
  - Freely and openly distributed

Cocke

New market segments

New opportunities

(and, of course, some new design and implementation constraints)

Processor and system segmentation

- Consumer desktops
- Business desktops
- Thin clients
- Workstations
- Servers
- Network processors
- Embedded and special purpose processors and systems
- Mobile processors and handheld devices

A few examples of new market segments

Thin clients

- Also called the Network Computer, Hollow PC, and the Internet Box
- Isn't a thin client just the old "dumb terminal" revisited?
  - Is it a step forward, a step backward, or marketing hype?
- Observation: The term "thin client" embraces a range of systems aimed at growing new consumer mass markets based on access to the Internet

Some thin client possibilities

- PDA, palmtop, smart card, and “wallet” versions
- TV set-top box versions with cable modem
- Video-game console-based versions
- Kiosk touch-screen versions (special purpose and general purpose)
- ISDN-phone versions
- Intelligent Appliances for e-Commerce applications
- WEB-terminals

Thin clients (cont.)

(from handheld devices to mobile computers)

- Many thin clients will include
  - Voice, data, and net access (Internet, intranet, and/or VPN)
    - What μArch support will enhance the acquisition, transmission, storage, and use of this type of information for these types of devices?
  - Support for wireless communications technologies
    - What μArch support will enhance the support and performance of specific wireless technologies?

Estimated market growth rates

- Communications chip market estimated to be 25% per year for the next several years
- Microprocessor chip market estimated to be 15% per year for the next several years
- Recent prediction by Andy Grove: By 2005 the demands of e-Commerce and new services would increase the world's processing requirements by 20-fold!
- Can/Should communication processing cores be included on-chip?

Why do microarchitects require an understanding of hardware, software, and systems?

- Because they've got to know what to do with the silicon and they've got to meet specific performance targets!
  - How to use the additional gates as we transition from 10M to 100M gates/chip
- What does the software really do?
  - Applications, languages, compilers, operating systems, new emerging markets
- How can the microarchitecture support it?
  - Instruction level parallelism, processing traces, multithread support, distributed/parallel computing support

Thin clients: moving computation from the client to the server

- Minimal storage and a small operating system
- No monitor (simple LCD screen), no floppies, and a flash disk or small form factor hard disk
- Bottom-line: For many thin client platforms, storage and processing are done on servers. Such thin clients:
  - Are dependent on the net to be useful
  - Shift work from the client to the server

Some multi-billion $ questions

- Who will the thin client users be?
- What will they be doing with these platforms?
- What is the business model for these newly emerging and somewhat undefined markets?

What will users be doing with thin clients?

- What applications will they be running?
- What are the run-time characteristics of these applications?
- Will an interpretive language (read that "processor independent" language) be used to execute simple applets on the client?
- Is any special graphics, multimedia, or communications support required?

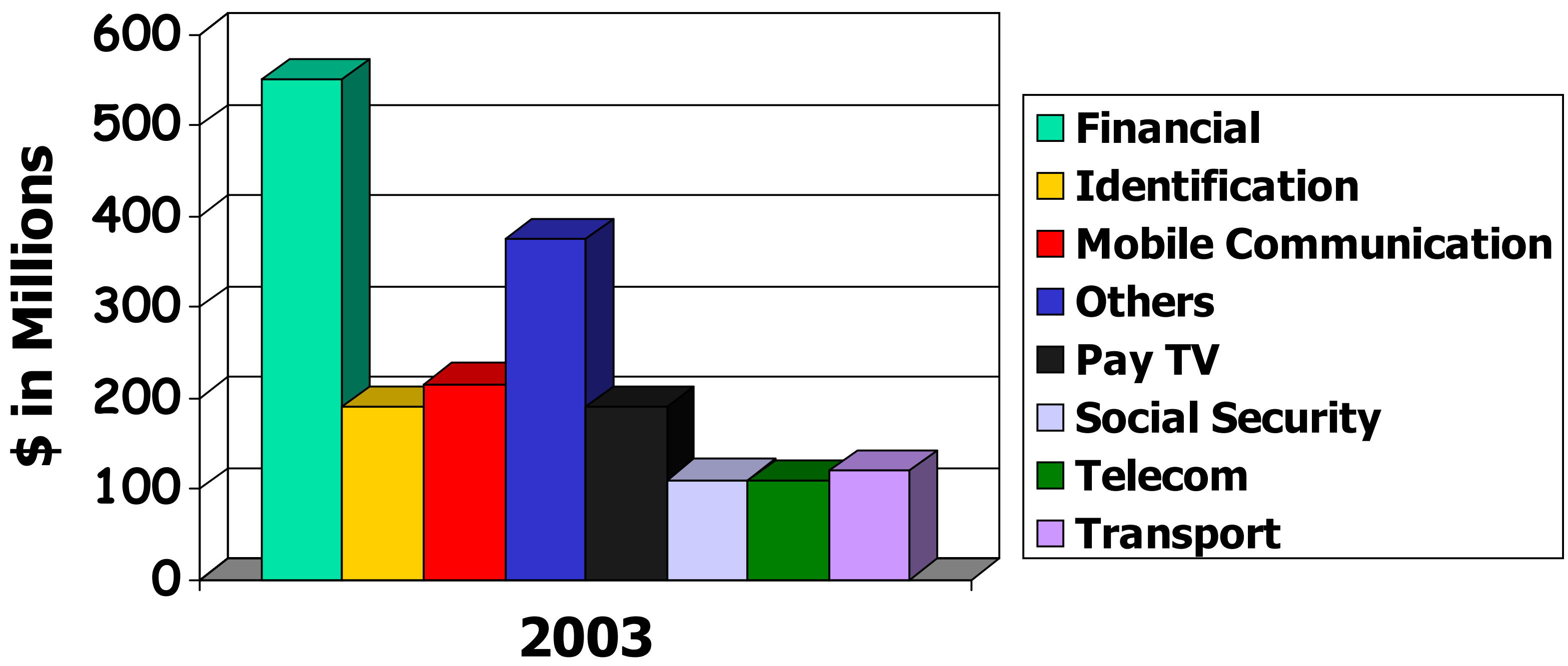
What will users be doing with thin clients? (cont.)

- What will the operating system be? Does it make a difference given the "platform independence" of WEB-technology?
- What should the µArch of the microprocessor be?
- Are any special features required to support the OS/application run-time environment?
- Does the µArch make a difference given the platform independence of WEB-technology?

Smart “smart cards”

- It’s been projected that by 2005, smart cards will be available that contain
  - 1-MByte of memory
  - 32-bit high-performance microprocessor
- Time to market of new platforms and applications will be based on reusable cores (both hardware cores and software cores)
  - E.g, Java engines and cryptography engines
  - E.g., off-the-shelf frameworks and components

Projected Applications



Source: Infineon Technologies (1999)

Will the µArch make a difference?

- Support for
  - Multimedia applications?
  - Multiple graphics interfaces?
- Certifiably secure?
- Will smart cards have their own internet addresses?
- Support for the MASSC requirements (multi-application secure smart cards)?

Microprocessors and microprocessor-based systems

- In addition to achieving higher and higher performance at lower and lower ASPs, microprocessors and microprocessor-based systems must be proven to be rather that conjectured to be
  - Reliable
  - Available
  - Scalable
  - Serviceable
  - Trusted

How can the microarchitect impact these important issues?

Back to the future
Core technologies that are important in the design and implementation of future classes of platforms.

The core technologies that I identify are based, in part, on the underlying assumption that an interdisciplinary team is involved in the specification, design, and implementation efforts.

Core Technologies I

- Algorithm development, analysis, and implementation
- Optimizing compiler technology
- Microkernel operating system design and implementation
- Processor and system co-design (vs. hardware and software co-design)
- Integrated simulation, prototyping, trace analysis, and performance measurement systems

Core Technologies II

- High-performance I/0 (buses, memory, and cache organizations), peripherals (e.g., graphics), and communications technologies
- Self-modifying & dynamically reconfigurable systems technology
- Consistent and enforceable international intellectual property treaties and statutes
  - A methodology that treats all design related material as digital assets within the framework of a digital asset management system

Core Technologies III

- Human/computer interaction technology
  - Making the computer fade into the background
- Educational institutions as deeply committed to education as they are to research
- Flexible and dynamic corporate cultures and organizational structures
- A body of knowledge dealing with the impact on and use of computer-based technology both by individuals and within organizations

Closing observations

- Successful organizations
  - Often change what they do and/or how they go about doing what they do in order to remain successful
  - That find it difficult to predict and incorporate major technological shifts central to their products and services stand a significant chance of withering away
- I believe that organizations involved in μArch and microprocessor R&D that have strengths in a number of the identified core technologies have a much higher probability of being successful than those that do not.

Closing observations (cont.)

- Industry and academia must collaborate closely in evolving the tools and environments that we use to design, implement, test, and study microprocessors and microprocessor-based systems
  - Unfortunately, because of the time and cost to develop the tools and environments, this is not always the case
- Consumer products and embedded systems are driving large parts of the computer industry today. We've been caught before in the trap of believing that we knew what customers need. Will we be caught again?

Questions?