10x10 must replace 90/10

The Future of Computer Architecture

Andrew A. Chien
April 27, 2010
Salishan Conference

*Views of the author alone....
Agenda

- 25 Years of Microprocessors
- Key Limits
- Traditional 90/10 Architecture
- 10x10 Architecture: A new Paradigm
- Observations
- Discussion
25 Years of Microprocessors (‘85 – ’10)

- Epochs of micro-architectures
  - Basic Micro (Intel 8080); Pipelines + caches, “risc shift” (MIPS/Sparc)
  - Superscalar (Power), Floating Point (...happened along the way)
    - uOps (dynamic translation) (Intel PPro)
  - Mmedia extensions (variable precision, vector) (Dec, Intel)
  - 64-bit (DEC, AMD); Multicore (all)

- 1000x performance!

© Andrew A. Chien, 2010

Die Area
- Integer Performance (X)
- FP Performance (X)
- Int Performance/Watt (X)

On-die Cache, Pipelined
Super-Scalar
OOO-Speculative
Deep Pipeline
Back to non-deep pipeline
The Transition to Multicore

- AMD Phenom (4 cores)
- Transistors (Thousands)
- Parallel App Performance
- Sequential App Performance
- Frequency (MHz)
- Number of Cores
- Typical Power (Watts)

Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond

(Courtesy K. Keutzer & D. Patterson)
Key Limits for Designs

1985-2005
- Transistor Count
- Design Complexity
- Valid. & Test Complexity
- Clock Rate
- Energy

2005-2020?
- Transistor Count
- Design Complexity
- Valid. & Test Complexity
- Clock Rate
- Energy
Traditional Optimization: 90/10 Rule

- Workloads: analyze and derive common cases (90%)
- Invent arch features, implementation optimizations with broad impact (90%)
- Improve performance by adding optimizations

Workloads

Abstracted “common” cases

“ILP”
“reuse locality”
“linear access”
“bit-field opns”
“branch patterns”

Optimizations

“pipelining”
“superscalar”
“caches & blocks”
“mmedia”
“branch pred”

© Andrew A. Chien, 2010
We need a NEW paradigm.

• The central element of the 90/10 paradigm is a focus on the 90 percent case
  o Broad focus on application workloads, derive common cases
  o Broadly applicable architecture improvements
  o Broadly applicable implementation improvements

• Improvements are aggregated into the general-purpose hardware “the Core”
  o Eliminates opportunities to customize for performance
  o Eliminates opportunities to customize to reduce power/energy
90/10 Aggregation leads to Energy Inefficiency

- Microprocessor complexity grows with each new optimization/feature
  - Instruction set compatibility (binaries must run)
  - Performance monotonicity (apps must not slow down)
  - New functionality – double fp, mmedia, virtualization, crypto, etc.
Approach #1: Integrate Lots of Traditional Cores

- Mainstream processors market approach (preserve software compatibility and performance monotonicity)
- IBM Power, Intel Xeon, AMD Opteron are delivering complex, high performance cores into the mainstream market and scaling the number of cores

Won’t get too far, low energy efficiency limits scaling
Approach #2: Simplify for Energy Efficiency, then Scaleup

- **Simplify**: strip out micro-architecture techniques; adding execution restrictions (SIMD, hierarchical control, restricted data access)
- **Examples**: Blue Gene/L, Blue Gene/P, Sequoia
  - And GPUs...
- **Caution**: Interconnect energy scaling
Customization improves Performance and Energy Efficiency

Microprocessors

General Purpose Microprocessors

Various DSP Products

MPEG2 Encrypt

MPEG2 802.11a

H.264

DSPs

100x

10x

MOPS/mW

Source: Intel

Accelerators can achieve 100x higher performance/watt
An Alternate Vision for Energy Efficient Computing

- Create nodes that are a collection of customized, special tools which are designed to purpose
  - A set of tools, each high performance and energy efficient for its computational structure
  - Tools complement each other in purpose and use (different specialties)
- Only one tool used at a time! (others powered off)
  - That tool achieves high performance and energy efficiency…
Remember: Key Limits for Designs

2005-2020?
- Transistor Count
- Design Complexity
- Valid.&Test Complexity
- Clock Rate
- Energy

2020+
- Transistor Count
- Design Complexity
- Valid.&Test Complexity
- Clock Rate
- Energy

Multiple accelerators can replace a general-purpose unit and increase performance and energy efficiency.
10x10 Optimization Paradigm

- Identify 10 most important applications; compute structures; datatypes (focus on 10 distinct bins)
- Make each 10x lower energy and 10x faster (100x more energy efficient) (optimize design and implementation separately)
- Compose together sharing memory hierarchy and interconnect (preserve the benefits of customization)
Examples of Accelerators

- Intel Polaris (80 core teraflop) and Core 2 Duo
  - 1M transistors, 1W/10GF vs. 100M, 100W/10GF => 100x perf/watt
- Grape DR N-body simulator vs. Conventional systems
  - 1000x perf/watt (or more!)
- Green Flash vs Opteron’s
  - 3MW vs 180MW for climate simulation (same speed), 60x energy efficiency
- Anton – molecular dynamics accelerator
  - 120x faster than fastest conventional implementation; even greater EE advantage

...
10x10 Microprocessor Architecture

- Many cores, each is 10 distinct accelerators achieving 100x better energy efficiency
  - 10x lower power enables 10x more cores
  - 10x better application performance on a core delivers 100x better overall performance
- Energy is the key limit, 10x10 approaches will outperform traditional
  - Produce highest performance per “core” (optimized implementation)
  - Produce highest performance chip (lowest energy/ops)
  - Product highest compute density (driven by lowest energy/ops)
10x10 is a paradigm shift

- Focus on 10pct parts of the workload (hooray customize for ME)
  - DON’T focus on the 90pct cases

- Build an energy-efficient, custom computing element (accelerator)
  - DON’T focus on the “general purpose” – though clearly we can throw one in

- Compose these together in energy efficient fashion
  - DON’T integrate them together deeply, as that will reduce their efficiency (performance and energy)

Claim: 10x10 microprocessors will be the dominant approach by 2020
Observations about 10x10

- 10 is arbitrary, but its >>1 to avoid the “gravity well” of the mainstream core
- The 10 slices should be designed complement each other and cover the space well.
  - Makes covering benchmarks suites and workloads easier!
- Different “customized” microprocessors might select 10 different slices / accelerators
- Is 10x10 for mainstream or HPC? Likely Both. Which first?
- Does 10x10 make sense before 2020? Yes.
10x10 Challenges (Future talks)

• Are there really 10 “dominant” modes of computation? And how to choose and exploit?

• How does software deal with 10x10? (too many targets)
  o Compilers
  o Applications
Summary

• Energy-limited scaling era; community is driving 90/10 optimization, cutting out 20 years of architecture
• Must focus on 10x10 – a new paradigm that will enable 100x improvement in energy efficiency
• Major Challenges –
  o Candidates for the 10, Selecting which 10; efficient architecture and implementation, efficient composition, software
• Let’s get to work on these challenges!
• Contact: andrew.chien@alum.mit.edu
What is the future of computing?

OR

OR

OR
Backup
Why 10by10 is not SoC

• SoC’s are macro-compositions of designs to meet a particular SYSTEM application need
  o Blocks operate concurrently; pipeline, etc.

• 10by10 is a micro-composition within a single computational building block
  o 10 elements designed for synergy
  o Need to understand the decomposition to design the 10
  o Selection of which 10 could perhaps be based on application (a la Green Flash customization)
Why 10x10 can Work: Transistors are Free

- Design Space, 65W, 100mm²

<table>
<thead>
<tr>
<th>Year</th>
<th>Logic MT</th>
<th>Cache MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>50</td>
<td>6/36</td>
</tr>
<tr>
<td>2014</td>
<td>100</td>
<td>25/150</td>
</tr>
<tr>
<td>2018</td>
<td>150</td>
<td>80/480</td>
</tr>
</tbody>
</table>

© Andrew A. Chien, 2010