Joint Architecture and Circuit Techniques to Address Process and Voltage Variability

Gu-Yeon Wei & David Brooks

School of Engineering and Applied Sciences
Harvard University
The Great Wall of Collaboration

Architect

Circuit Designer
The Great Wall of Collaboration

Architect

Circuit Designer
The Great Wall of Collaboration

Architect

Circuit Designer
Architecture & Circuits Groups

Not shown:
- Andrew
- Krishna
- Ruwan
- Ben
Collaborative Projects

• SW+Arch+HW for efficient power delivery
  – Understanding Voltage Variations in CMPs Using a Distributed Power Delivery Network (DATE ’07)
  – Toward a SW Approach to Mitigate Voltage Emergencies (ISLPED’07)
  – DeCoR: A Delayed-Commit and Rollback Mechanism for Handling Inductive Noise in Processors (HPCA’08)
  – System-Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators (ASGI’07, HPCA’08)

• SW+Arch+HW to combat process variations
  – Mitigating the Impact of Process Variation on CPU RF and Execution Units (MICRO’06)
  – Process Variation Tolerant 3T1D-Based Cache Architectures (ASGI’07, MICRO’07)
  – A Process Variation Tolerant FPU with Voltage Interpolation and Variable Latency (ISSCC’08)
Today’s Topics

• System-Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators
  – Wonyoung Kim, Meeta Gupta, Wei and Brooks
  – To be presented at HPCA in Feb. 2008

• Process Variation Tolerant 3T1D-Based Cache Architectures
  – Xiaoyao (Alex) Liang, Ramon Canal (UPC Barcelona), Gu-Yeon Wei and David Brooks
  – To be presented at MICRO in Dec. 2007
Seminar Part 1

SYSTEM-LEVEL ANALYSIS OF FAST, PER-CORE DVFS USING ON-CHIP SWITCHING REGULATORS
Voltage Variability Movie

3 cores running *bzip*, 1 core idle

1 core running *bzip*, 3 cores idle
Motivating Example

- Can we move the off-chip regulator onto the processors?
- If yes, WHY?
Supply Noise Comparison

 resonance

 BW limitation of on-chip regulator
Fast DVFS

- Off-chip regulators limited to microsecond-scale transitions
- On-chip regulators enable nanosecond-scale voltage transitions
  - Can we leverage this fast switching?
Outline

• Motivation
• Offline DVFS
• On-chip regulator design
• Simulation analysis
• Summary & future work
Fast DVFS w/ On-Chip Regulators

Questions to answer:
1. Does fast DVFS offer power savings?
2. For CMPs, do we want one global supply or per-core voltage control?
3. What does an on-chip regulator cost us?
4. How can architecture help regulator design?
5. How does this all add up?
DVFS Overview

• Minimize energy consumption w/ bounded performance loss
  – Exploit CPU slack from asynchronous memory events
    (i.e., L2 miss) to reduce frequency (F) and voltage (V)

• Offline DVFS control
  – Formulate as integer linear programming (ILP) optimization problem
  – Oracle uses memory vs. CPU boundedness to set V/F across different windowed intervals
    • 4 V/F settings assumed
  – Compare different intervals (100ns to 100μs)
**DVFS Architecture Study**

<table>
<thead>
<tr>
<th>Frequency</th>
<th>1GHz @ 65nm</th>
<th>Vdd</th>
<th>1 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core area</td>
<td>16mm²</td>
<td>Fetch/Issue/Retire</td>
<td>2/2/2</td>
</tr>
<tr>
<td>Branch Penalty</td>
<td>7 cycles</td>
<td>Branch Predictor</td>
<td>BTB (1K entries)</td>
</tr>
<tr>
<td>Hybrid Branch Predictor</td>
<td></td>
<td>RAS (32 entries)</td>
<td></td>
</tr>
<tr>
<td>Int registers</td>
<td>32</td>
<td>FP registers</td>
<td>32</td>
</tr>
<tr>
<td>IL1</td>
<td>32KB, 32-way, 32B block</td>
<td>DL1</td>
<td>32KB, 32-way, 32B block</td>
</tr>
<tr>
<td>Hit/miss latency</td>
<td>2/1 cycles</td>
<td>Hit/miss latency: 2/1 cycles</td>
<td></td>
</tr>
<tr>
<td>MESI protocol</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITLB entries</td>
<td>64</td>
<td>DTLB entries</td>
<td>128</td>
</tr>
<tr>
<td>MSHR size</td>
<td>8</td>
<td>Write Buffer size</td>
<td>16</td>
</tr>
<tr>
<td>L2 size</td>
<td>512 KB</td>
<td>L2 miss penalty</td>
<td>200 cycles</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>(Memory Cycles/Total Runtime)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ocean-con</td>
<td>Large Scale ocean simulation</td>
<td>0.47</td>
</tr>
<tr>
<td>fft</td>
<td>Fast Fourier Transform</td>
<td>0.40</td>
</tr>
<tr>
<td>facerec</td>
<td>CSU Face Recognizer</td>
<td>0.22</td>
</tr>
<tr>
<td>cholesky</td>
<td>Cholesky factorization</td>
<td>0.197</td>
</tr>
<tr>
<td>raytrace</td>
<td>Tachyon Ray Tracer</td>
<td>0.058</td>
</tr>
<tr>
<td>mcf-mcf-mcf-mcf</td>
<td>4-high memory-bound</td>
<td>0.697</td>
</tr>
<tr>
<td>mcf-mcf-mcf-applu</td>
<td>3-high memory-bound(mcf) and 1-high cpu-bound (applu)</td>
<td>0.697(mcf) and 0.051(applu)</td>
</tr>
<tr>
<td>mcf-applu-applu-applu</td>
<td>2-high memory-bound(mcf) and 2-high cpu-bound (applu)</td>
<td>0.697(mcf), 0.051(applu)</td>
</tr>
<tr>
<td>applu-applu-applu-applu</td>
<td>1-high memory-bound(mcf) and 3-high cpu-bound (applu)</td>
<td>0.697(mcf), 0.051(applu)</td>
</tr>
<tr>
<td>applu-applu-applu-applu</td>
<td>4-high cpu-bound(applu)</td>
<td>0.051</td>
</tr>
</tbody>
</table>

- **Processor model**
  - 4 simple Xscale-like in-order cores
  - Private L1, shared L2
- **Simulation framework**
  - SESC multi-core simulator
  - Wattch power modeling
  - Cacti cache simulator
  - Orion
  - MESI-based cache coherence
  - Multithreaded and multi-programming benchmarks
Ocean’s DVFS Opportunities

- Multithreaded ocean running on all 4 cores exhibits variable activity between cores
- Per-core voltage again offers more DVFS opportunities
**fft’s DVFS Opportunities**

- Multithreaded *fft* running on all 4 cores exhibits variable activity between cores
- Per-core voltage offers more DVFS opportunities
Benefits of Fine-Grained DVFS

- Off-chip regulator $\rightarrow$ 100$\mu$s – static (app-level) intervals
  - OS-level DVFS control
- On-chip regulator $\rightarrow$ 100ns – 1$\mu$s intervals
  - Needs online DVFS control
Global vs. Per-Core DVFS (multithreaded applications)

- DVFS interval = 100ns
- Per-core DVFS offers more savings
- Savings vs. benchmark trend tracks “variability”
Global vs. Per-Core DVFS
(multi-programming applications)

- DVFS interval = 100ns
- \textit{mcf} = memory-bound app; \textit{applu} = CPU-bound app
- Power savings for mix of memory- and CPU-bound apps
Regulator Design

Conventional buck converter w/ hysteretic control

- $P_{\text{delivered}} = \frac{1}{2} LI^2 F_{\text{switching}}$
- On-chip multiphase buck converter
  - Higher $F_{\text{switching}}$
  - Smaller $L$ & $C$
  - Lower $V_{\text{ripple}}$ and/or smaller filter $C$

Multi-phase buck converter
Power Delivery Options

(x4 for per-core DVFS)

- Can we leverage architecture to reduce the droop?
Current Staggering

- Burn power to reduce voltage droop
Voltage Transition Overhead

- Scale up voltage before increasing frequency
- Drop frequency before decreasing voltage
- Power overhead = area between curves
Regulator Specifications

- Optimized \( F_{\text{switching}} \) with respect to losses
  - Balance DVFS overhead with regulator loss

<table>
<thead>
<tr>
<th></th>
<th>off chip</th>
<th>off-chip &amp; on-chip</th>
<th>off-chip &amp; on-chip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One Power Domain</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of phases for on-chip regulator</td>
<td>8</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>On-chip regulator switching frequency (MHz)</td>
<td>170</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Decoupling capacitance (nF)</td>
<td>40</td>
<td>40</td>
<td>10 per core (total: 40)</td>
</tr>
<tr>
<td>Voltage margin (%)</td>
<td></td>
<td></td>
<td>±10</td>
</tr>
</tbody>
</table>
Energy Breakdown Comparison

The diagram illustrates the energy breakdown comparison for different configurations of CPU core supply regulators. The configurations include:

1. One Off-Chip Regulator (100μs DVFS interval)
2. One On-Chip Regulator (100ns DVFS interval)
3. Four On-Chip Regulators (100ns DVFS interval)

The chart shows the energy consumption (% of Processor Energy with no DVFS) across various applications, with different colors representing Processor Energy, Staggering Overhead, DVFS Overhead, On-Chip Regulator Loss, and Off-Chip Regulator Loss.

The applications illustrated on the x-axis are specific to the context of the diagram and are typically related to different tasks or workloads that vary in energy consumption.

The y-axis represents the energy consumption in percentage of the total processor energy with no DVFS intervention.

This comparison helps in understanding the impact of different regulation strategies on energy efficiency and can guide in making informed decisions for system design and optimization.
Relative Energy Savings
Putting It All Together

• Energy savings with fast DVFS offset by
  – On-chip regulator loss
  – Voltage transition power overhead
  – Current staggering overhead

• Per-core DVFS attractive for CMP systems
  – Must consider scalability of on-chip regulators

• Next steps:
  – Meeta is investigating fast DVFS scaling algorithms to leverage fast, fine-grained voltage switching
  – Wonyoung is designing the regulator
Seminar Part 2

PROCESS VARIATION TOLERANT 3T1D-BASED CACHE ARCHITECTURES
• As Moore’s Law continues and on-chip dimensions get smaller, imperfections in the fabrication process affect device performance more and more…
• Past: Worried about wafer-to-wafer, chip-to-chip variations
• Now: Worry about within-die, transistor-to-transistor variations

(Source: Friedberg, SPIE’06)  (Source: K. Bernstein, IBM J.R&D’06)
Variability Trends

• In the past...
  - wafer to wafer
  - chip to chip
  - core to core

• Now...
  - block to block
  - array to array
  - transistor to transistor
On-Chip Memory

• On-chip memory is a huge fraction of die area
SRAM scaling: A Tale of Two Conferences?

- Is SRAM scaling slowing down?
- Plots include circuit techniques to improve reliability (e.g., dual voltage, boosted WL, etc.)
Problems with 6T

- Susceptibility to process variations (PV)
  - Performance variations (Read/Write delay variations)
  - Bit flips due to voltage noise and leakage
  - Stuck at faults b/c too much mismatch

![Diagram of 6T memory cell](image-url)
Dealing with variability in memories

• Microarchitectural techniques
  – Traditional ideas to deal with soft errors
    • Parity or ECC
    • Cache scrubbing
  – PVT-induced soft errors much more frequent than radiation-induced soft errors
    • Must understand the system-level issues

• What’s the problem?
  – Fighting or feedback
    • Sensitive to mismatch
      → Boosted array or wordline voltage?
  – Bitline leakage
    • Large variations in leakage currents
      → Shorter bitlines?
Data Usage in L1

- On average, 90% of data accessed in first 6K cycles
Proposed Solution

- Use 3T1D dynamic cells to replace 6T cells

- Why?
  - Higher immunity to process variations
  - Absorb delay variation into cell “retention time”
  - No inherent fighting → no bit flips
  - Lower power (leakage and dynamic)
  - Higher density possible

- But what about refresh?
  - Use architectural insights and techniques to deal with dynamic data storage

- Where?
  - Analyzed register files (RF) and L1 data caches
  - eDRAMs being considered for L2 caches and above…
What is a 3T1D cell?

- Gated-diode selectively boosts stored data ("1") during reads
- Non-destructive reads allows for column multiplexing
• What retention time is “good enough”?
Simulation Setup

• Baseline: 4-wide Out-of-order machine
  – 20FO4 pipelines
  – 80-entry RF
  – 64KB, 4-way set-associative I- and D-caches
• sim-alpha simulator used to calculate instructions per cycle (IPC)
• 8 SPEC2000 benchmarks
Variation Model

- Monte Carlo analysis of process variation impact on memory cell delay and power
  - 32nm PTM, Vdd = 1V
  - Considered typical and extreme PV scenarios
  - Correlations based on Friedberg’s chip measurements

<table>
<thead>
<tr>
<th></th>
<th>Typical</th>
<th>Severe</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{L/L_{\text{nominal}}} \text{ (WID)}$</td>
<td>5%</td>
<td>7%</td>
</tr>
<tr>
<td>$\sigma_{V_{th}/V_{th}} \text{ (WID)}$</td>
<td>10%</td>
<td>15%</td>
</tr>
<tr>
<td>$\sigma_{L/L_{\text{nominal}}} \text{ (D2D)}$</td>
<td>5%</td>
<td>5%</td>
</tr>
</tbody>
</table>
Cache Configuration

• 64KB cache
  – 4-way Set Associative, 512b cache lines
  – 2 Read/1 Write ports
  – 8 256x256 subarrays
  – 64 Sense Amps per subarray

Three Stage Cache Access

| Address Generation Decode | WL Drive Bitline Swing Sense Amp | Output Logic Drive |
Cache Data Array Floorplan

- 256x256 Subarray
- Column Mux: 256
- Sense Amps: 64
- Way Mux (Late Select from Tag)
- Way Select (From Tag Array)
- Data Out: 64
- Row and Column Decode
- 256x256 Subarray

Wei & Brooks
IEEE Denver Chapter Technical Seminar (11/1/07)
Global Refresh Scheme

- 8 cycles to refresh one cache line (SA-limited)
- 2K cycles to refresh entire cache (476ns @ 4.3GHz)
- ~6µs retention time (no variations)
- Refresh takes 8% of cache bandwidth
- IPC hit < 1%
6T Performance under typical variations

![Bar chart showing chip probability against normalized frequency (performance)]
3T Performance under typical variations

![Graph showing performance and cache retention time for 3T1D and worst-case benchmark with dynamic power over cache retention time.]

- **Performance**
  - Mean perf. for 3T1D
  - Perf. for 3T1D (worst-case benchmark)
  - 500ns

- **Dynamic power**
  - Normal operation dyn. pwr. for 3T1D
  - Refresh dyn. pwr. for 3T1D
  - Total dyn. pwr. for 3T1D
Three chips under severe variations
Line-Level Schemes: Refresh Policies

- Refresh Policies
  - Full-refresh: Per-line counter forces refresh when needed
  - No-refresh: Rely on L2 inclusion properties
  - Partial-refresh: Threshold counter chooses one of the two policies
Line-Level Schemes: Replacement Policies

• Replacement Policies
  – Dead-sensitive Placement
    • Avoid using “dead” lines when performing placement
  – Retention-sensitive placement (RSP-FIFO)
    • Order lines in descending retention time
    • New lines are assigned the longest retention time line (and old ones reshuffle)
  – Retention-sensitive placement (RSP-LRU)
    • MRU block is assigned the longest retention time
Evaluating Policies

Performance

no refresh partial refresh full refresh no refresh partial refresh full refresh RSP-FIFO RSP-LRU

LRU DSP

good chip median chip bad chip

Wei & Brooks
IEEE Denver Chapter Technical Seminar (11/1/07)
Pushing policies to the limits

1. 65nm, typical, 1.1V
2. 45nm, typical, 1.1V
3. 32nm, typical, 1.1V
4. 32nm, severe, 1.1V
5. 32nm, typical, 0.9V
6. 32nm, severe, 0.9V
Power Analysis (Dynamic)

- Refresh power is small (~10% overhead for better schemes)
Power Analysis (Leakage)

- Substantial leakage savings

![Graph a. Cache leakage power distribution for 1X 6T](image1)

![Graph b. Cache leakage power distribution for 3T1D](image2)
Reliable Memory Summary

• Transient nature of data in L1 cache allows for architecturally-simple refresh schemes for 3T1D memories

• Provides PV-tolerant on-chip memories
  – Comparable performance to “ideal” 6T
  – Lower leakage power
  – Low HW overhead

• Similar results observed for 3T1D register files and instruction caches

• Test chip planned for fab in Spring 2008