Blackcomb: Hardware-Software Co-design for Non-Volatile Memory in Exascale Systems

Perspectives on Blackcomb Simulation Tools

Jeffrey Vetter, ORNL
Robert Schreiber, HP Labs
Trevor Mudge, University of Michigan
Yuan Xie, Penn State University

Presented to
DOE X-Stack PI Meeting
Boston
28 May 2014

http://ft.ornl.gov ◆ vetter@computer.org
Blackcomb: Hardware-Software Co-design for Non-Volatile Memory in Exascale Systems (started 2010)

http://ft.ornl.gov/trac/blackcomb

**Objectives**

- Rearchitect servers and clusters, using nonvolatile memory (NVM) to overcome resilience, energy, and performance walls in exascale computing:
  - Ultrafast checkpointing to nearby NVM
  - Reoptimize the memory hierarchy for exascale, using new memory technologies
  - Replace disk with fast, low-power NVM
  - Enhance resilience and energy efficiency
  - Provide added memory capacity

**Established and Emerging Memory Technologies – A Comparison**

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>eDRAM</th>
<th>NAND Flash</th>
<th>PCRAM</th>
<th>STT RAM</th>
<th>ReRAM (1T1R)</th>
<th>ReRAM (Xpoint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Retention</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Cell Size (F²)</td>
<td>50-200</td>
<td>4-6</td>
<td>19-26</td>
<td>2-5</td>
<td>4-10</td>
<td>8-40</td>
<td>6-20</td>
<td>1-4</td>
</tr>
<tr>
<td>Read Time (ns)</td>
<td>&lt; 1</td>
<td>30</td>
<td>5</td>
<td>10⁴</td>
<td>10-50</td>
<td>10</td>
<td>5-10</td>
<td>50</td>
</tr>
<tr>
<td>Write Time (ns)</td>
<td>&lt; 1</td>
<td>50</td>
<td>5</td>
<td>10⁵</td>
<td>100-300</td>
<td>5-20</td>
<td>5-10</td>
<td>10-100</td>
</tr>
<tr>
<td>Number of Rewrites</td>
<td>10¹⁶</td>
<td>10¹⁶</td>
<td>10¹⁶</td>
<td>10⁴-10⁵</td>
<td>10⁸-10¹²</td>
<td>10¹⁵</td>
<td>10⁸-10¹²</td>
<td>10⁶-10¹⁰</td>
</tr>
<tr>
<td>Read Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Write Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Power (other than R/W)</td>
<td>Leakage</td>
<td>Refresh</td>
<td>Refresh</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Sneak</td>
</tr>
</tbody>
</table>
Blackcomb Use Cases
Device Level Cell and Array Modeling

Fig. 1. Schematic view of 0T1R and 1D1R ReRAM structure: array structure of 0T1R (a) and 1D1R (b); circuit diagram of 0T1R (c) and 1D1R (d).

Fig. 4. First stage design flow. ($V_{\text{worst}}, V_{th}$: calculated value and constraint of worst case voltage. $\Delta I_{rd}, \Delta I_{th}$: calculated value and constraint of read noise margin. $M, N$: number of wordline and bitline.)

**Tradeoffs in Exascale Memory Architectures**

- Understanding the tradeoffs
  - ECC type, row buffers, DRAM physical page size, bitline length, etc

Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications

- **Problem**
  - Do specific memory workload characteristics of scientific apps map well onto NVRAMs’ features?
  - Can NVRAM be used as a solution for future Exascale systems?

- **Solution**
  - Develop a binary instrumentation tool to investigate memory access patterns related to NVRAM
  - Study realistic DOE applications (Nek5000, S3D, CAM and GTC) at fine granularity

- **Impact**
  - Identify large amount of commonly existing data structures that can be placed in NVRAM to save energy
  - Identify many NVRAM-friendly memory access patterns in DOE applications
  - Received attention from both vendor and apps teams

Rethinking Algorithm-Based Fault Tolerance

- Algorithm-based fault tolerance (ABFT) has many attractive characteristics
  - Can reduce or even eliminate the expensive periodic checkpoint/rollback
  - Can bring negligible performance loss when deployed in large scale
  - No modifications from architecture and system software

- However
  - ABFT is completely opaque to any underlying hardware resilience mechanisms
  - These hardware resilience mechanisms are also unaware of ABFT
  - Some data structures are over-protected by ABFT and hardware

Evaluation

- We use four ABFT (FT-DGEMM, FT-Cholesky, FT-CG and FT-HPL)

- We save up to 25% for system energy (and up to 40% for dynamic memory energy) with up to 18% performance improvement
Simulation Tools
### Prediction Techniques Ranked

<table>
<thead>
<tr>
<th>Technique</th>
<th>Speed</th>
<th>Ease</th>
<th>Flexibility</th>
<th>Accuracy</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-hoc Analytical Models</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Structured Analytical Models</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Aspen</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Simulation – Functional</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Simulation – Cycle Accurate</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Hardware Emulation (FPGA)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Similar hardware measurement</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Node Prototype</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Prototype at Scale</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Final System</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Architectural Simulators

• Sniper, McSim, Zsim, Marss, Gem5 etc.
  – These simulators model processor components, e.g. cache, memory, core
  – They provide stats on cache/memory access, their overlap, hit rate, etc.
  – We have used McSim and Sniper

• NVRAM cache parameters are derived from device level simulators like NVSim and imported into Sniper, McSim, etc
## A quick, partial overview of architectural simulators - Mar 2014

<table>
<thead>
<tr>
<th>Simulator</th>
<th>CPU-GPU Hybrid</th>
<th>Detailed Processor Modeling</th>
<th>Detailed Main Memory Modeling</th>
<th>Uses Pin?</th>
<th>Coherence</th>
<th>x86 support</th>
<th>Full-System</th>
<th>Parallelization</th>
</tr>
</thead>
<tbody>
<tr>
<td>McSimA+</td>
<td>No</td>
<td>?</td>
<td>Yes</td>
<td>Yes</td>
<td>?</td>
<td>x86-64</td>
<td>No</td>
<td>?</td>
</tr>
<tr>
<td>Multi2Sim</td>
<td>Yes</td>
<td>Yes (OoO)</td>
<td>Yes</td>
<td>No</td>
<td>?</td>
<td>x86</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MacSim</td>
<td>Yes</td>
<td>?</td>
<td>Yes (DRAMsim)</td>
<td>Yes</td>
<td>?</td>
<td>x86</td>
<td>No</td>
<td>?</td>
</tr>
<tr>
<td>GEMS</td>
<td>No</td>
<td>Yes (OoO)</td>
<td>No</td>
<td>Simics</td>
<td>Yes</td>
<td>Yes (x86, sparc ...)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Gem5</td>
<td>No</td>
<td>Yes (OoO, inorder ...)</td>
<td>Yes (DRAMsim)</td>
<td>No</td>
<td>Yes (GEMS)</td>
<td>x86</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Marss</td>
<td>No</td>
<td>Yes</td>
<td>Yes (DRAMsim)</td>
<td>Qemu</td>
<td>Yes</td>
<td>x86-64</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sniper</td>
<td>No</td>
<td>No (suitable only for uncore)</td>
<td>No</td>
<td>Yes</td>
<td>?</td>
<td>x86-64</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>GPGPUsim</td>
<td>No</td>
<td>?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Gem5-GPU X86</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (DRAMsim)</td>
<td>No</td>
<td>Yes(GEMS)</td>
<td>X86-64</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Gem5-GPU Alpha</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (DRAMsim)</td>
<td>No</td>
<td>Yes(GEMS)</td>
<td>Alpha</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ESESC</td>
<td>Yes</td>
<td>Yes (OoO, inorder)</td>
<td>?</td>
<td>Qemu</td>
<td>Yes</td>
<td>No (Arm, MIPS)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Graphite</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>?</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Zsim</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Architectural Memory Simulators

- **DRAMsim2**
  - a cycle accurate DRAM system simulator at the architectural level

- **NVmain**
  - a cycle accurate main memory simulator to simulate NVM at the architectural level

- Often integrated with processor-simulators, such as McSim, Zsim, Gem5, Marss etc.

- **FPGA Emulation for Memory Behaviors**
  - Using FPGA connected to simulator running on host to investigated complex, high throughput memory behaviors
Device-level Simulators

• CACTI (and its variants)
  – An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model
  – Primarily for SRAM, eDRAM and DRAM (2D and 3D)

• NVSim
  – Models the area, timing, dynamic energy and leakage power of NVM technologies
  – For STT-RAM, PCM, ReRAM (NVRAMs), NAND flash
  – Extension of PCRAMsim
Q & A

More info: vetter@computer.org
Contributors and Recent Sponsors

  - Publications: [https://ft.ornl.gov/publications](https://ft.ornl.gov/publications)

- Department of Energy Office of Science
  - Vancouver Project: [https://ft.ornl.gov/trac/vancouver](https://ft.ornl.gov/trac/vancouver)
  - Blackcomb Project: [https://ft.ornl.gov/trac/blackcomb](https://ft.ornl.gov/trac/blackcomb)
  - ExMatEx Codesign Center: [http://codesign.lanl.gov](http://codesign.lanl.gov)
  - CS Efforts: [http://science.energy.gov/ascr/research/computer-science/](http://science.energy.gov/ascr/research/computer-science/)

- DOE 'Application' offices

- National Science Foundation Keeneland Project: [http://keeneland.gatech.edu](http://keeneland.gatech.edu)

- NVIDIA CUDA Center of Excellence at Georgia Tech

- Other sponsors
  - ORNL LDRD, NIH, AFRL, DoD
  - DARPA (HPCS, UHPC, AACE)
References

- http://nvsim.org/
- http://wiki.nvmain.org/
- http://wiki.umd.edu/DRAMSim2/
- http://scale.snu.ac.kr/mcsim.en.html
- http://snipersim.org/
**Integration**

- SST can be used with Gem5, which can be used with NVMain/DRAMsim2 (for memory) and CACTI/NVSim (for cache)