CSEP 548: Computer Systems Architecture

NVRAM

Luis Ceze, Spring 2017

Phase-change memory (PCM)

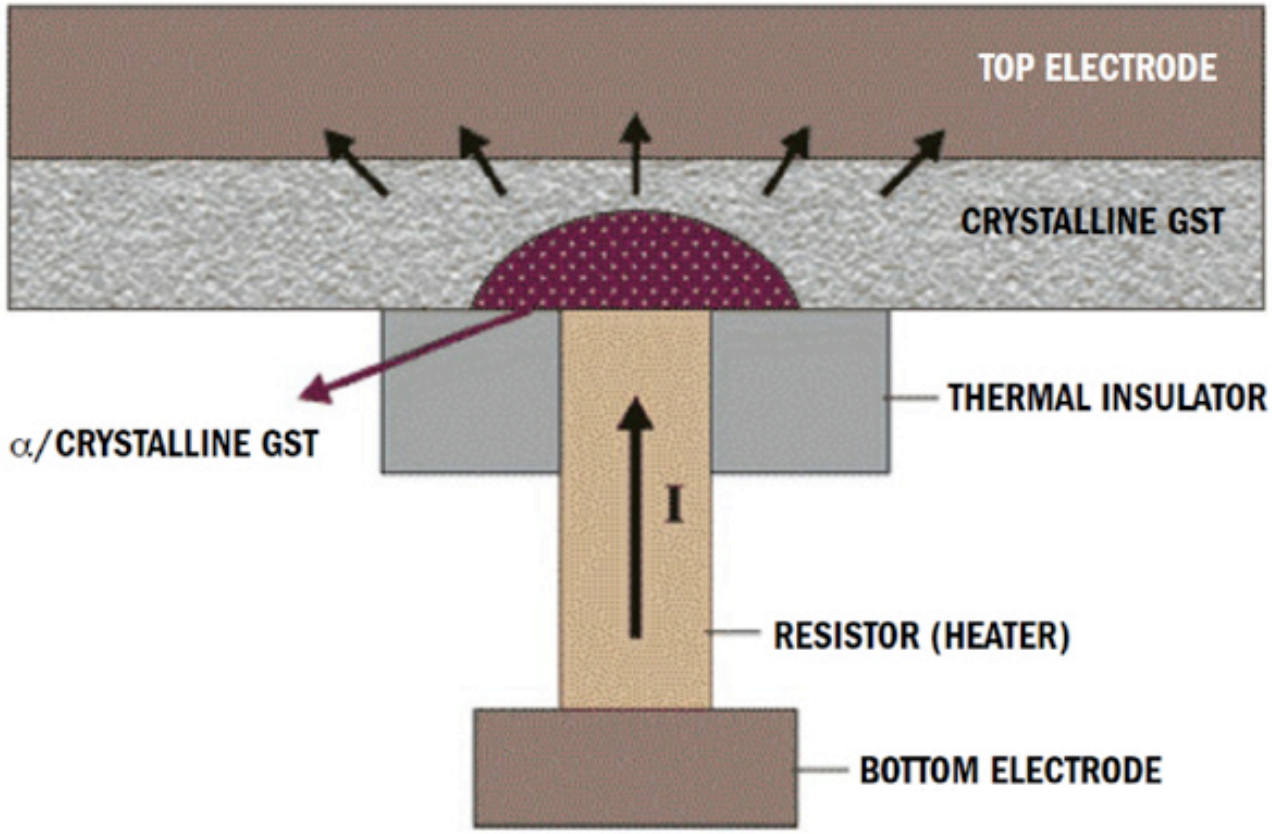


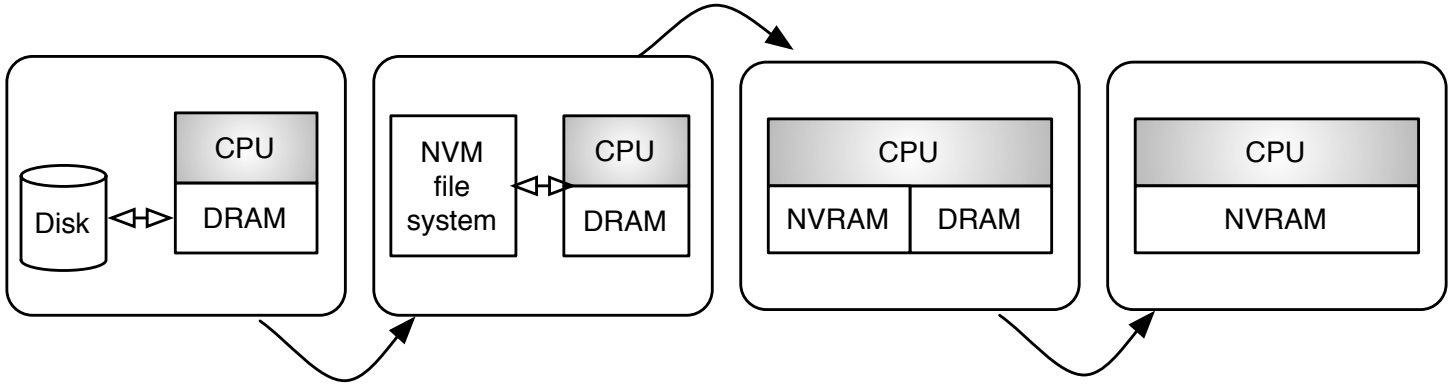
Figure 1. Typical PCM device structure.

Properties

- Fast reads (~2-3x slower than DRAM)
- Slower writes
- Endurance: 10^8 write cycles
- Byte-addressable

Canonical NVRAM uses

- Density
  - Use PCM for its density characteristics only
  - Topic of most architecture research
- Durability
  - Expose non-volatility characteristics to SW
  - Spectrum of HW/SW interface choices:



Implications on OS Design

- Assumptions (far out): Entire memory is fast, non-volatile (D)
- OS components
  - Virtual Memory
  - File systems
- Execution models
  - Application installation and launch
  - Software faults
  - Software updates
  - Recycling OS state/reboot
- System properties
  - Reliability
  - Security, Privacy and Forensics

OS Components

- Virtual Memory
  - No more paging/swapping?, page granularity?
  - Unify memory protect and file system access control?
  - Reconsider single-address-space designs?
- File systems
  - Most *straightforward* use of NVM
  - Should we *rethink FS interface* in this context?
    - Most past research on NVM-based FS keeps current interface
  - Perhaps have in-memory DB services?

Execution Models

- Applications and Processes
  - Does the concept of install and launch apply?
    - Always a “running” image (e.g., see iOS)
  - How to deal corrupted state during faults?
  - How is application code updated?
- OS operations
  - What does a reboot do?
  - Do we still need reboots?

System Properties

- Reliability
  - Data corruption
    - Dangling pointers might corrupt durable objects
    - Implicit data “sanitization” in two-level store
  - Data portability
    - Big deal that we can’t move data physically between systems?
- Security, Privacy and Forensics
  - Device theft worse?
  - Cold boot attacks worse?
  - Better forensics with NVM gave fast and frequent checkpoints?

New Uses for NVM

- New programming models
  - Expose durable data-structures more directly
- Continuous copy-on-write
  - Remember the entire execution history

- High-frequency power on/off
  - Better power proportionality

- <Anything else?>