



**Western Digital<sup>®</sup>**

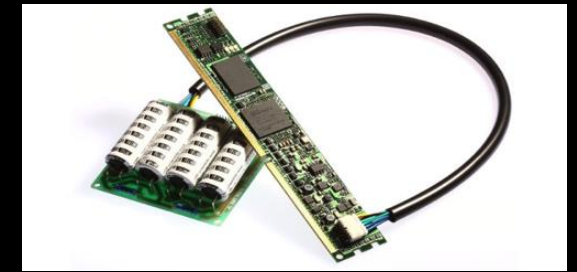
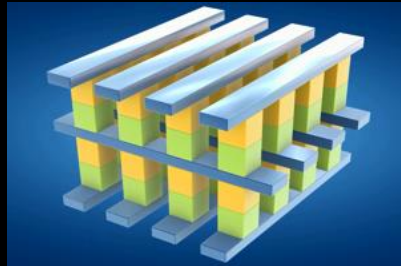
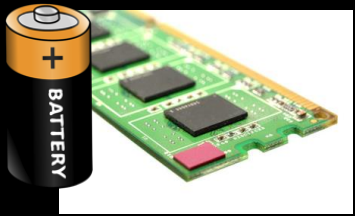
# **RISC-V Support for Persistent Memory Systems**

*Matheus Ogleari, Storage Architecture Engineer*

June 26, 2018

# Persistent Memory

- State-of-the-art, hybrid memory + storage properties
- Supported by hardware (NVRAM)



- Supported by software

## Persistent Memory Support in OS


### Using DAX in Windows

#### DAX Volume Creation

```
→ Format n: /dax /q
→ Format-Volume -DriveLetter n -IsDAX
```


#### DAX Volume Identification

```
Is it a DAX volume?
→ call GetVolumeInformation("C:\", ...)
→ check lpFileSystemFlags for FILE_DAX_VOLUME (0x20000000)
```



### NVM.PM.FILE ACTIONS

- Implemented in kernel
  - Map
  - Sync
- Implemented in userspace
  - Optimized\_flush
  - Optimized\_flush\_and\_verify



File system	Metadata atomicity	Data atomicity	Mmap Atomicity [1]
BPFS	Yes	Yes [2]	No
PMFS	Yes	No	No
Ext4-DAX	Yes	No	No
SCMFS	No	No	No
Aerie	Yes	No	No
NOVA	Yes	Yes	Yes

# Persistent Memory

- Also supported by some ISAs... but not RISC-V ☹️
- Relies on instructions and specific microarchitecture features

- Uncacheable logging



- Cache flushing



- Memory barriers



# RISC-V Changes

- New instructions
  - `ucst` `rs2`, `off(rs1)`
    - Uncacheable write to memory, works like a normal store but completely bypasses cache hierarchy
    - Could be multiple instructions to allow for word (`ucsw`), byte (`ucsb`), and half-word stores (`ucsh`)
  - `uclid` `rd`, `off(rs)`
    - Uncacheable read from memory, bypasses cache
    - May allow for word (`uclw`), byte (`uclb`), and half-word loads (`uclh`)

# RISC-V Changes

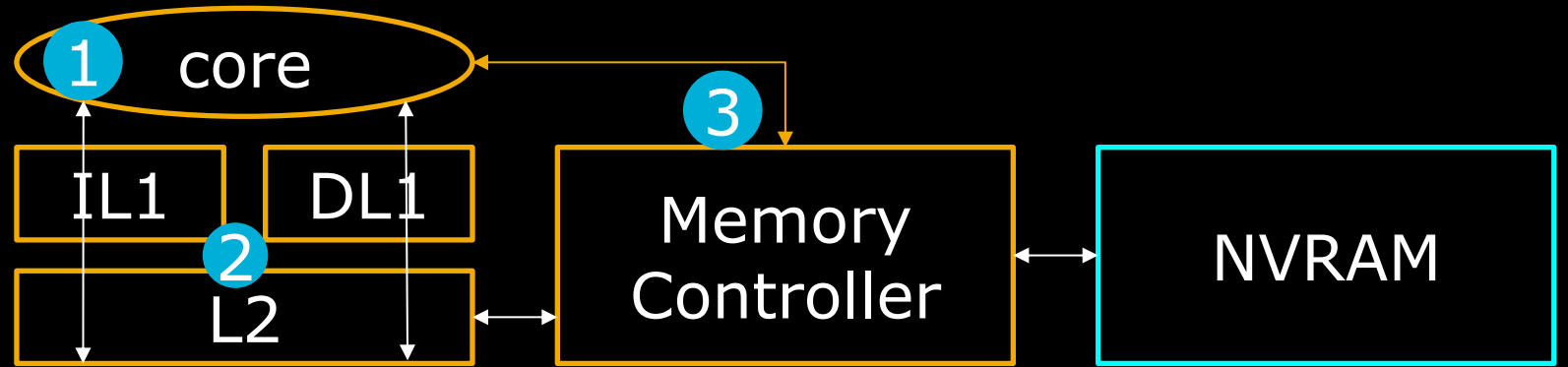
- New instructions
  - `sfence`
    - Memory barrier for store instructions
    - All stores before `sfence` must commit to memory before all stores after `sfence`
  - `clwb off(rs)`
    - Cache Line Write-Back
    - Forces cache to write-back cache line containing address specified by `rs` plus offset
    - Only occurs if cache line is dirty (resets this bit)
    - Cache line does not become invalidated

# Summary of Changes

1. Processor, ISA, decoder

2. Cache controller

3. Memory Controller



# Impact of Changes

- Research and academic impact
  - Software-based persistent memory work now has ISA tools to work with
  - Hardware-based persistent memory work now has the datapath in the microarchitecture to use for design implementation
  - Even if instructions are not used, the fundamental persistent memory functionality is present and enabled

# Applications

- WHISPER [Nalli et al, 2017]
  - Benchmark suite comprised of ten persistent memory applications
  - Currently cannot be run on RISC-V
  - Relies on pmemlib and pmemio support, libraries cannot compile in RISC-V
  - Most persistent memory work evaluate experiments with WHISPER
- Persistent Memory-Aware Systems
  - NOVA [Xu 2016] is a persistent memory-aware file system (pmfs) that uses this functionality, these changes would enable it to run on RISC-V
  - Crash consistency and recovery is necessary for proper persistent memory functionality [Pelley 2014]



# Applications

- Hardware persistent memory designs require basic microarchitecture support
  - Hardware Undo+Redo Logging [Ogleari 2018]
  - ATOM [Joshi 2017]
  - DudeTM [Liu 2017]
- RISC-V opens up hardware-driven research and complexity analysis
  - Would allow for direct study of hardware metrics like area, cycle time, and energy
  - Would allow for FPGA implementations of persistent memory systems
  - RTL design can be done for persistent memory blocks

# Our Ongoing Work

Persistent  
Memory Work



RISC-V



Rocket/BOOM  
Chip



Xilinx Zynq zc706 FPGA board



An abstract graphic on the left side of the slide, consisting of numerous thin, overlapping lines in shades of red, orange, yellow, and cyan. These lines flow and curve across the frame, creating a sense of motion and digital energy. The lines are most dense on the left and right sides, with some crossing in the center.

# Western Digital®