# Testing Controller Memory Buffer Performance In An NVM Express SSD

Ramyakanth Edupuganti, Senior Product Research Engineer,
Dr. Stephen Bates, Senior Technical Director,
Microsemi Corporation

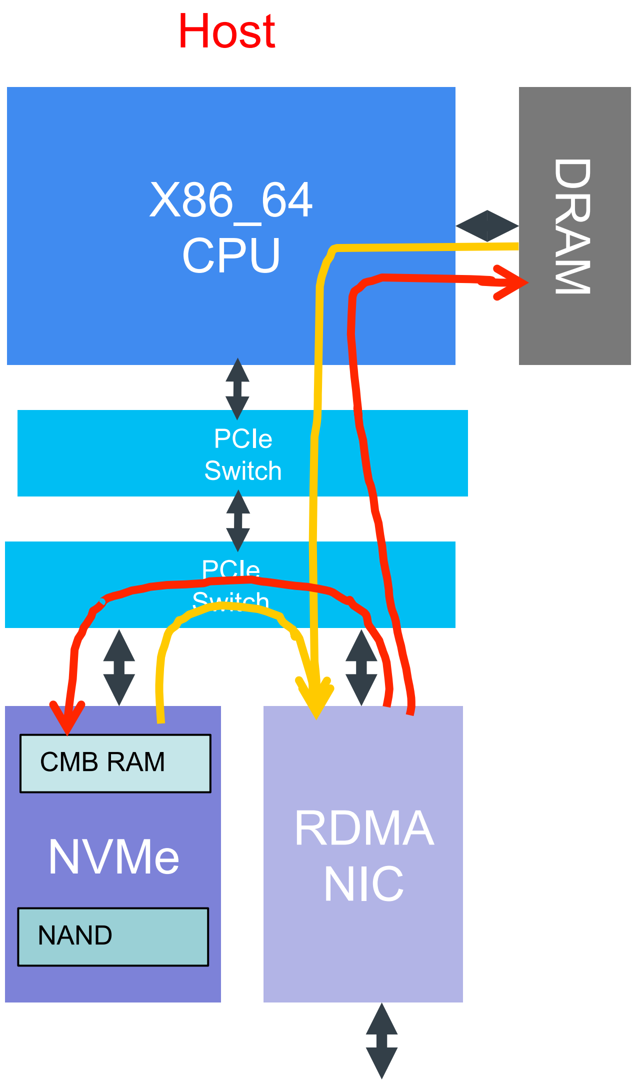
August 9th, 2016

# Overview

- Background
- Controller Memory Buffers
- High-Speed SSDs
- Enabling Linux Support
- Performance Results
- Conclusions

# Background

- PCIe network and storage devices can generate and consume several GB/s.
- Devices have either a high performance DMA engine, a number of exposed PCIe BARs or both.
- Until the Controller Memory Buffer, any high-performance transfer of information between two PCIe devices has required the use of a staging buffer in system memory.
- The bandwidth to system memory is not compromised when high-throughput transfers occurs between PCIe devices.

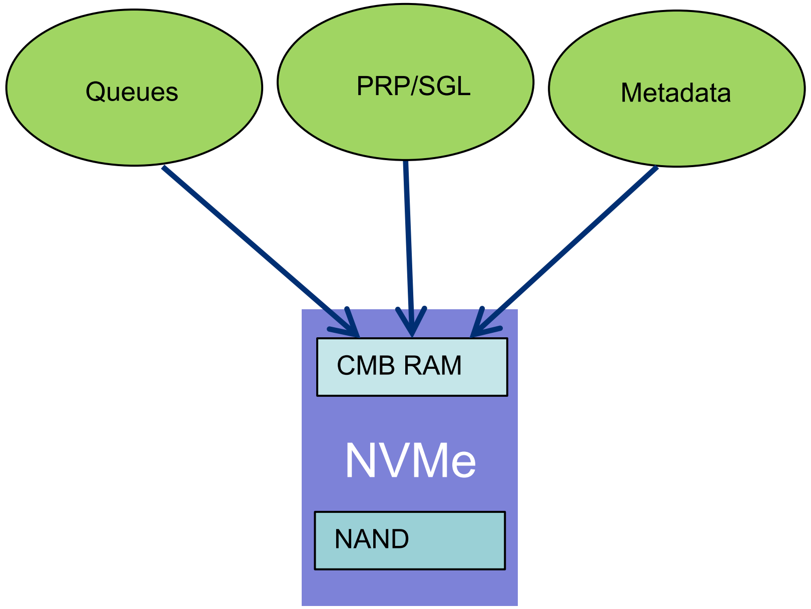


# Controller Memory Buffers

- Controller Memory Buffers (CMBs) were added to the NVM Express standard in revision 1.2

- CMBs are PCIe BARs (or regions within a BAR) that can be used to store either generic data or data associated with an NVMe block command

- With appropriate implementation, CMBs can be made persistent across power cycles (MRAM, poly-caps, 3DX-point, etc.,)



Flash Memory SUMMIT

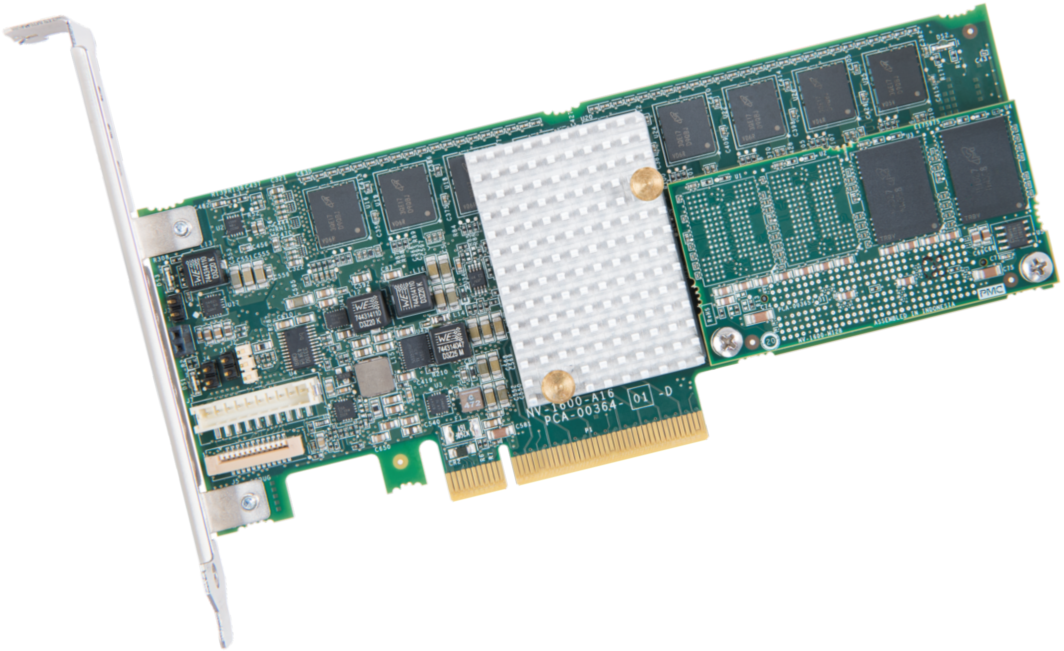
Microsemi

# High-Speed Solutions

- In Direct Memory Interface (DMI) mode, Microsemi Flashtec NVRAM cards achieve:

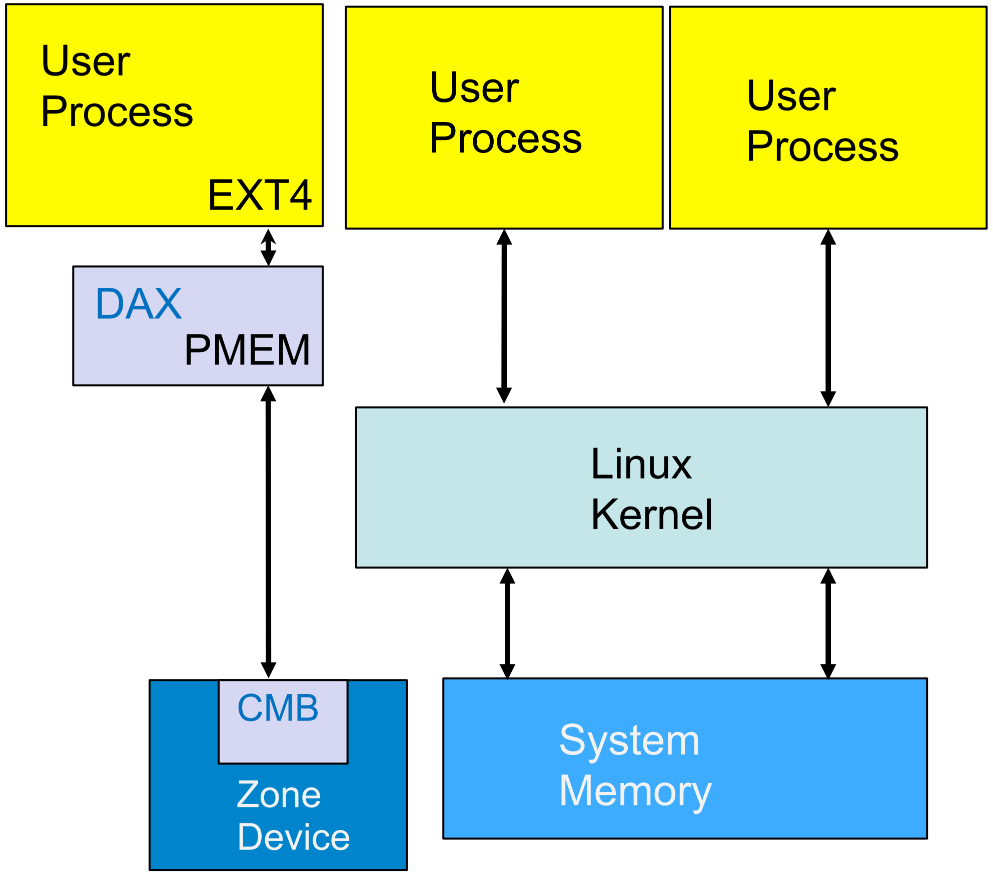
- Sub-microsecond latency for small access sizes

- 10 million IOPs for small access sizes



# Enabling Linux Support Features

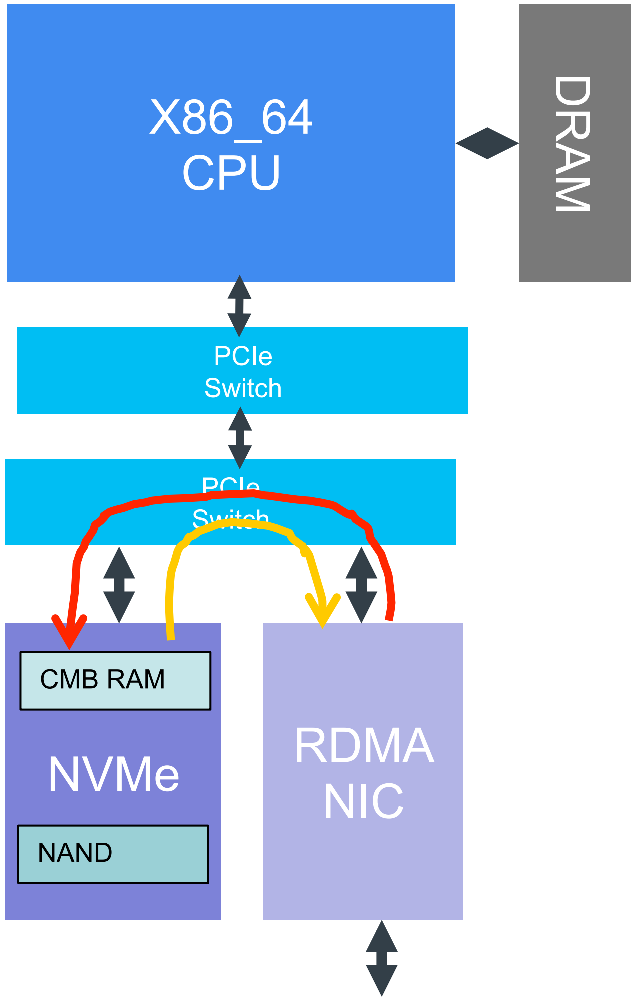
- Linux support has been enabled by several recent additions to the kernel, including:
- ZONE_DEVICE-the ability to associate a range of memory addresses (PFNs)
- PMEM-a ZONE_DEVICE device driver that exposes memory region to the rest of the OS
- DAX-a framework that allows a memory-addressable block device to bypass the page cache
- STRUCT PAGE SUPPORT-PMEM devices can optionally include struct page backing for DMA



# Performance Results

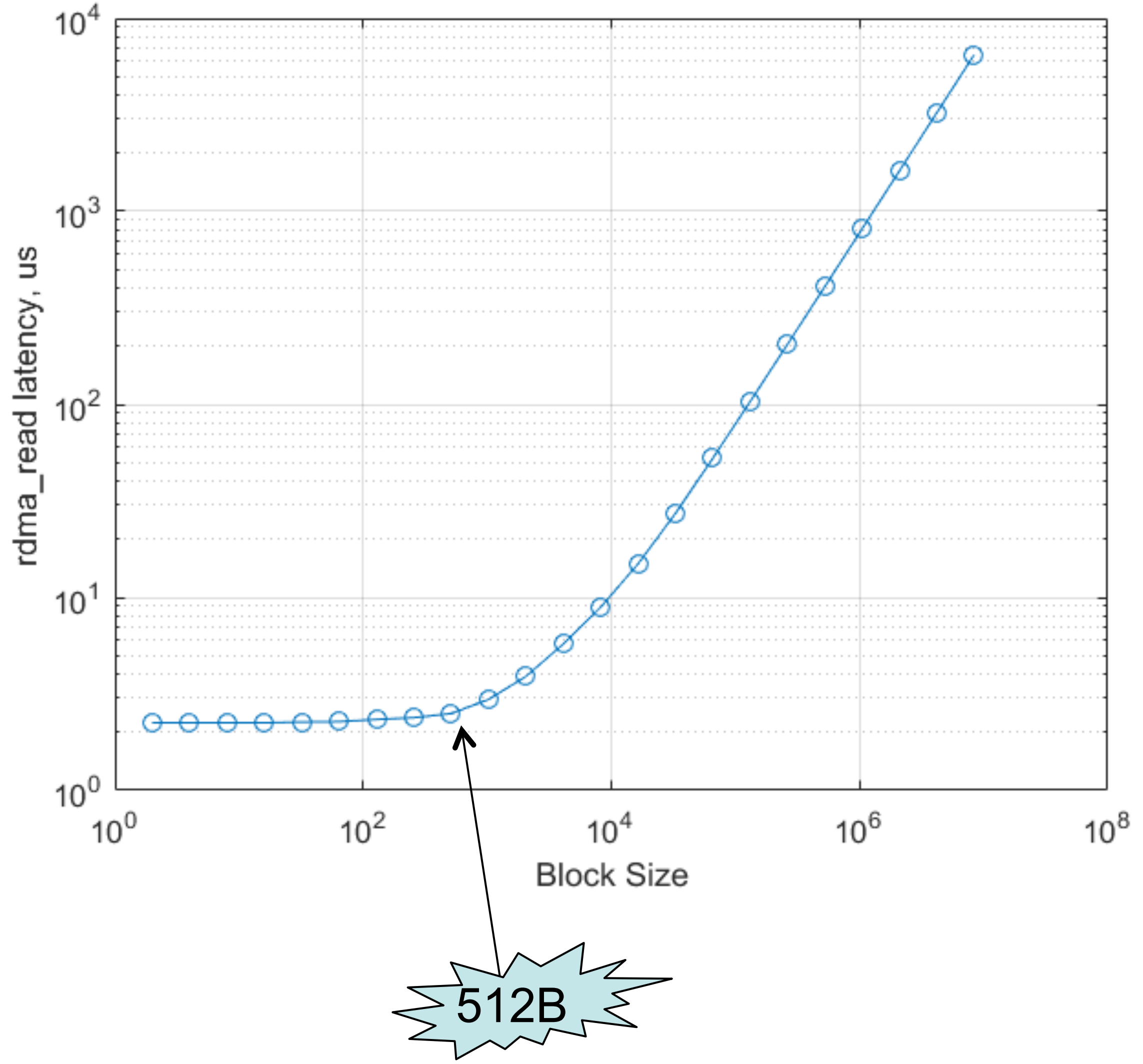
- The DMA engines in the RDMA device can target the BAR on the IOPMEM device and can either use the mmap() on the IOPMEM direct or mmap() files on a DAX-mounted filesystem as the RDMA memory regions.
- Performance:

| Write BW | Read BW |
|---|---|
| 4 GB/s | 1.2 GB/s |



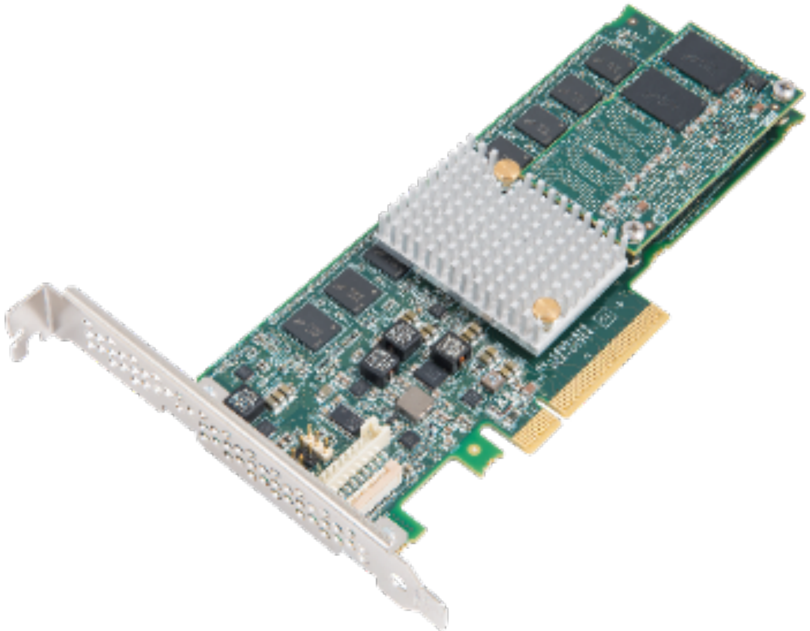
# Performance Results (continued)

- Sub 3 µs read times for small access sizes
- CX4, RoCE, and no RDMA switch
- Latency rises for larger accesses (these are unaligned and byte addressable)
- Results for 4 KB are around 6 µs. This is 6 µs total, not incremental!



# Conclusions

- The Linux kernel has been adapted in preparation for new NVMs and memory-attached NVM.
- We are building upon prior advancements in PCIE IO memory subsystem technology by adding new performance-enhancing features, and have enabled IO memory as a DMA target.
- Our example driver exposes IOMEM as both a DAX-enabled block device and an mmap()’-able region.
- We show good performance between PCIe devices (the datapath avoids CPU when a PCIe switch is used).



# THANKS!!

Microsemi

Come and visit us at booth #213

www.microsemi.com