Coherent Storage: the Brave New World of Non-Volatile Main Memory

Dejan Vucinic

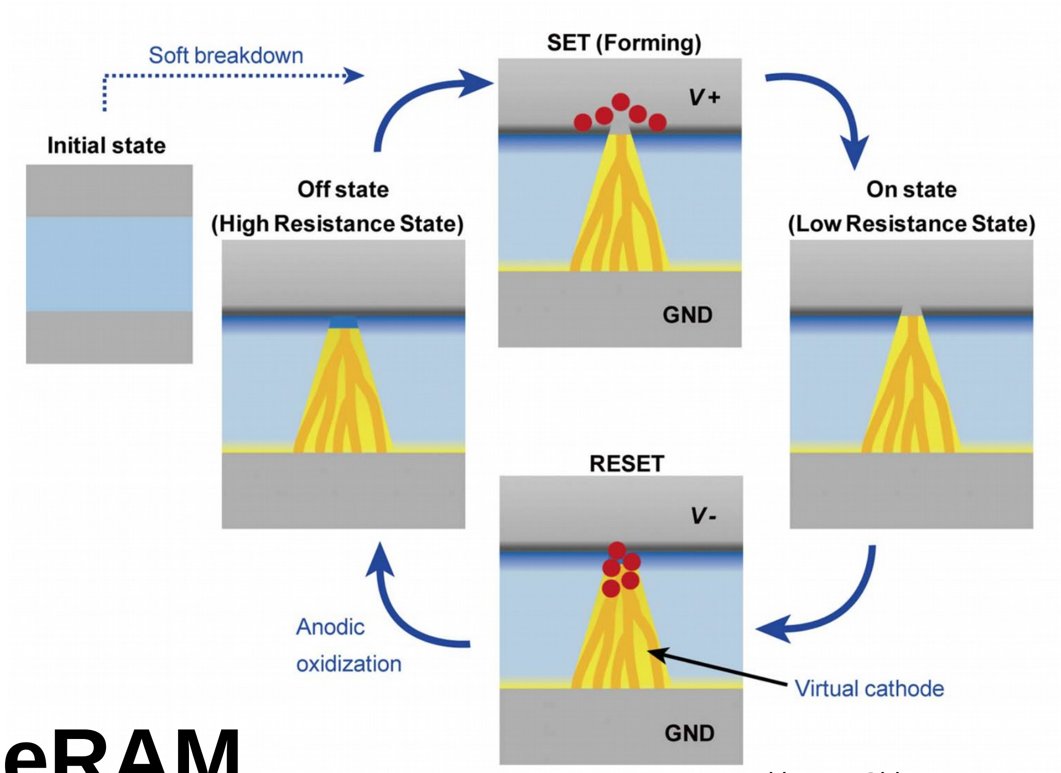
July 12th, 2016

Credits

- Zvonimir Bandic
- Kiran Gunnam
- Martin Lueker-Boden
- Luis Vittorio Cargnini
- Qingbo Wang
- Damien Le Moal
- Cyril Guyot
- Md Kamruzzaman
- Chao Sun
- Minghai Qin
- Luiz Franca-Neto
- Seung-Hwan Song
- Filip Blagojevic
- Robert Mateescu

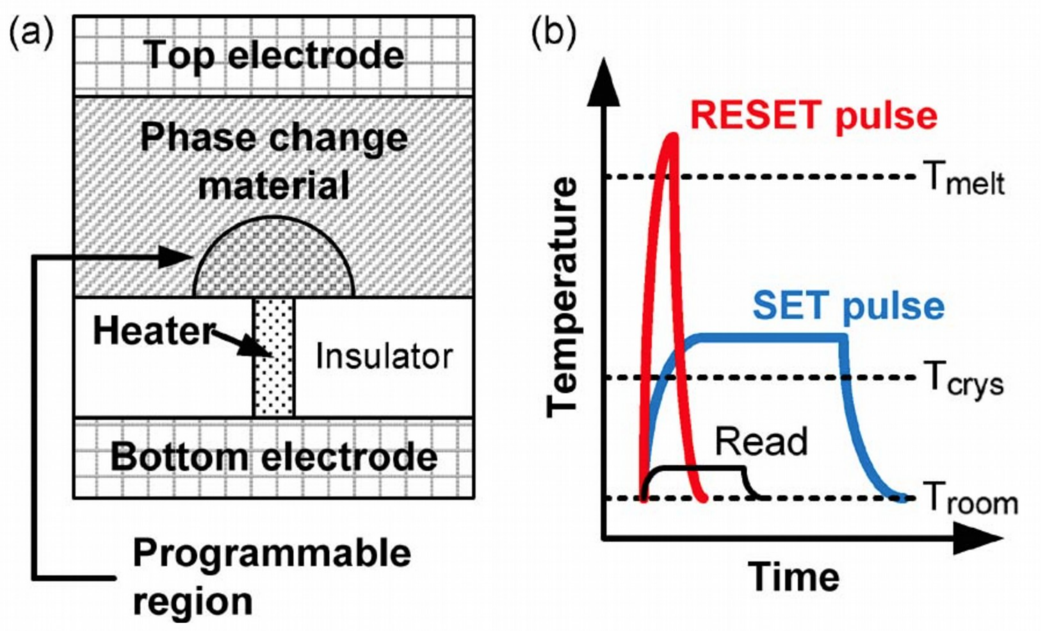
Emerging Resistive Non-Volatile Memories

PCM

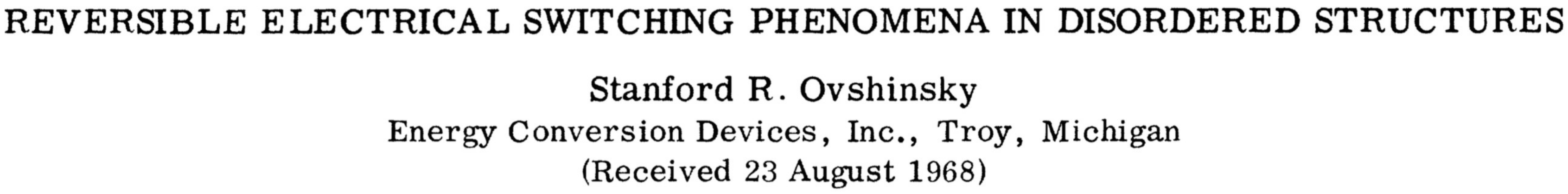


ReRAM

From Akinaga, Shima, Proc IEEE 2010



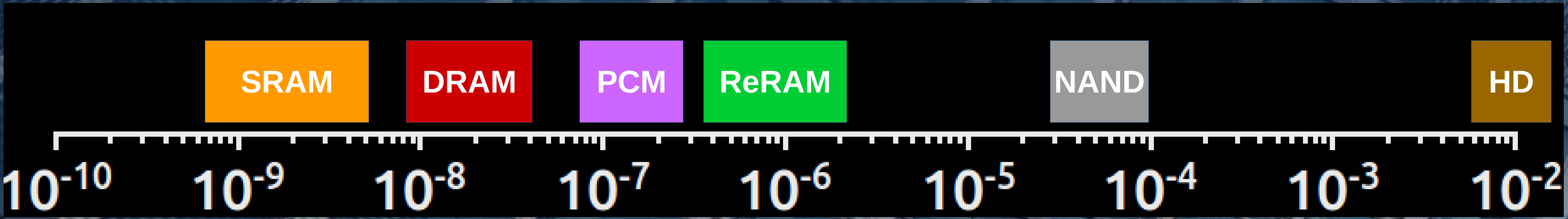
From H.-S. P. Wong et al. Proc IEEE 2010



REVERSIBLE ELECTRICAL SWITCHING PHENOMENA IN DISORDERED STRUCTURES

Stanford R. Ovshinsky
Energy Conversion Devices, Inc., Troy, Michigan
(Received 23 August 1968)

Read Latency of Emerging Resistive NVMs



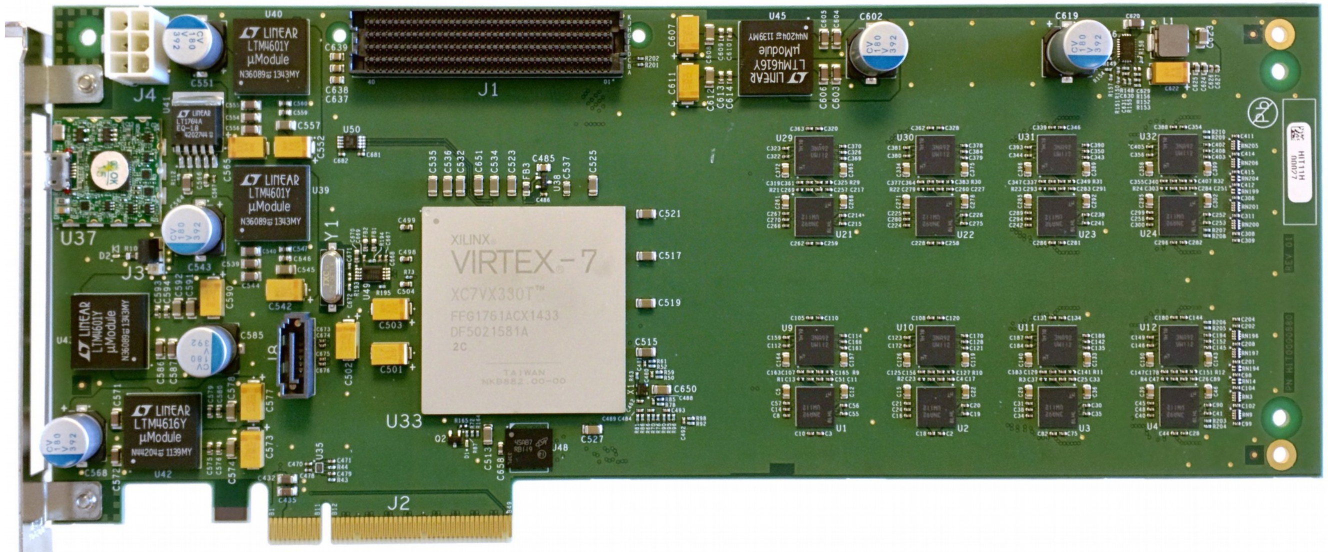
Where can we attach eNVMs?

Is it memory?

⛔

- Doesn't work today
- Major changes required to DDR protocol, controller IP

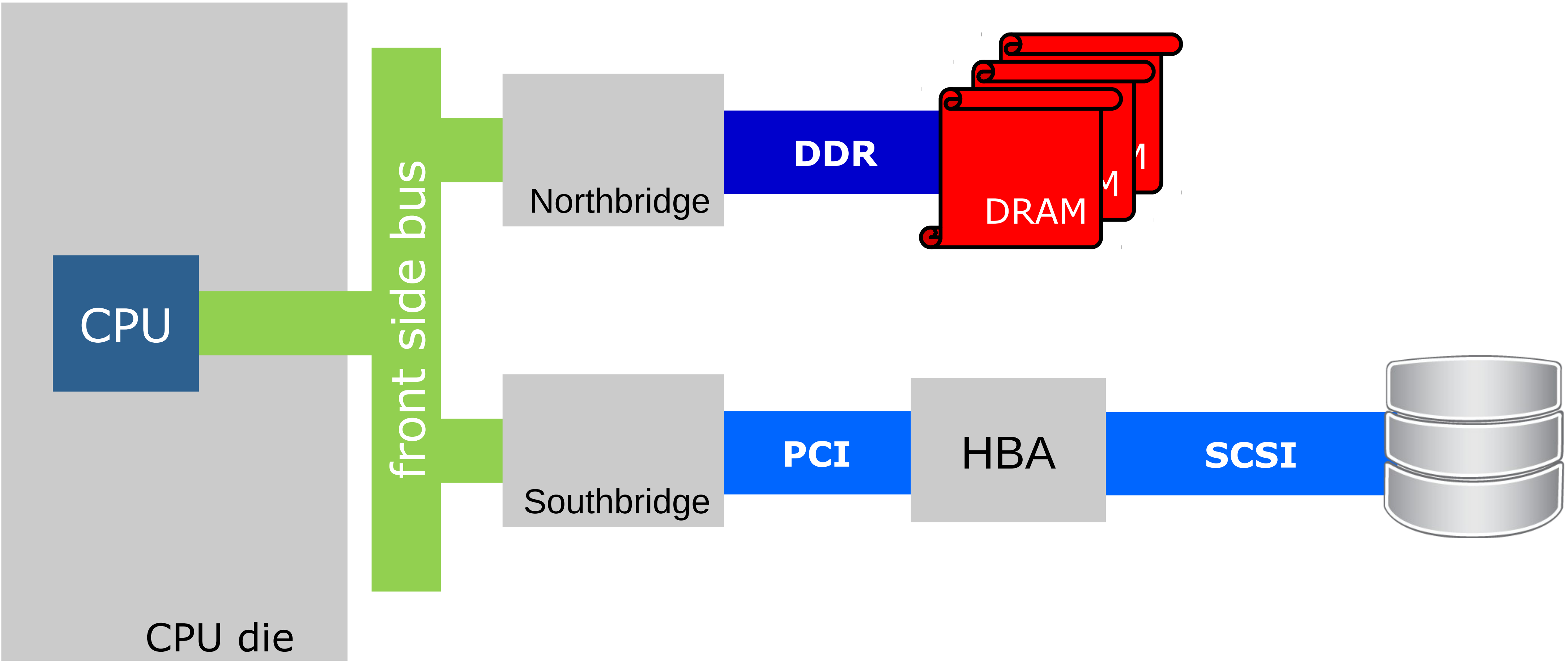
Is it storage?



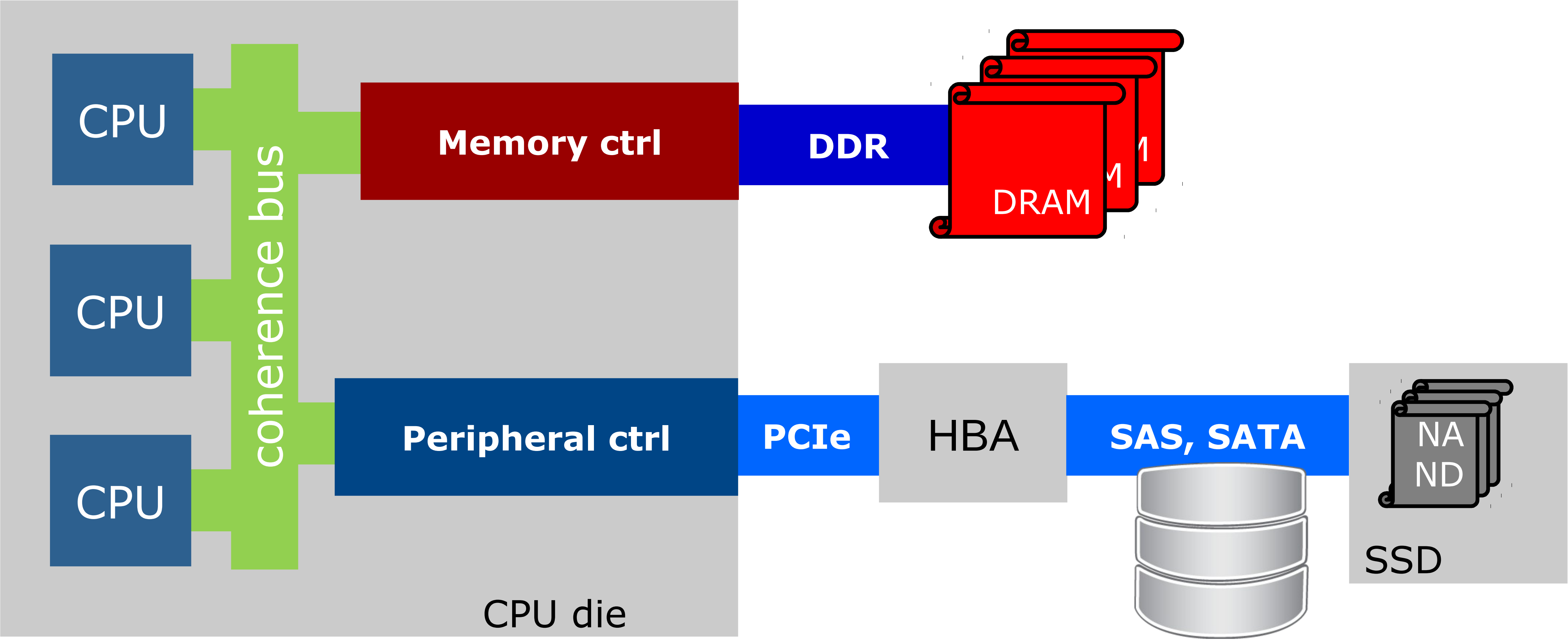
✓

- Works well today, but meh
- Latency of fast SSD dominated by PCIe latency—lost main advantage of resistive NVM

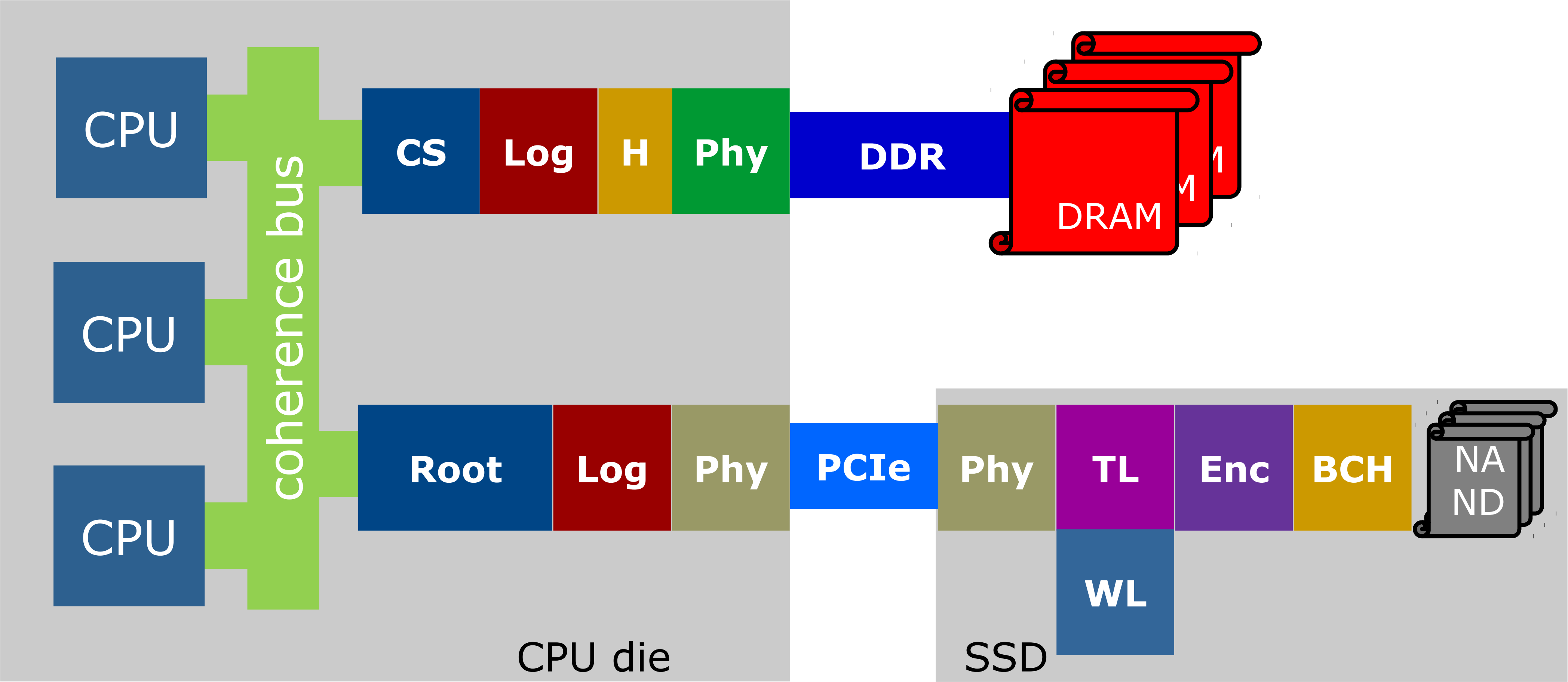
Once upon a time…



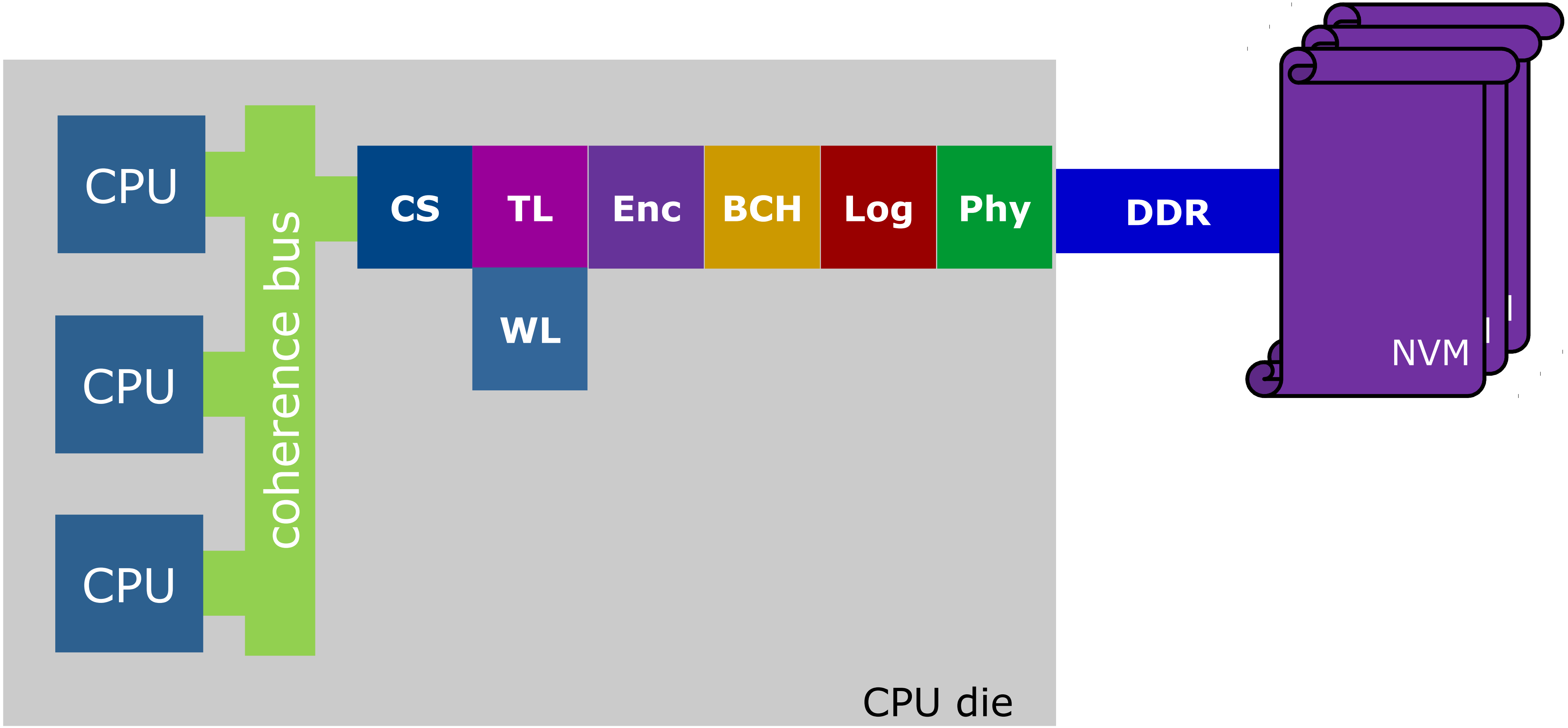
99.3% of servers today look like this



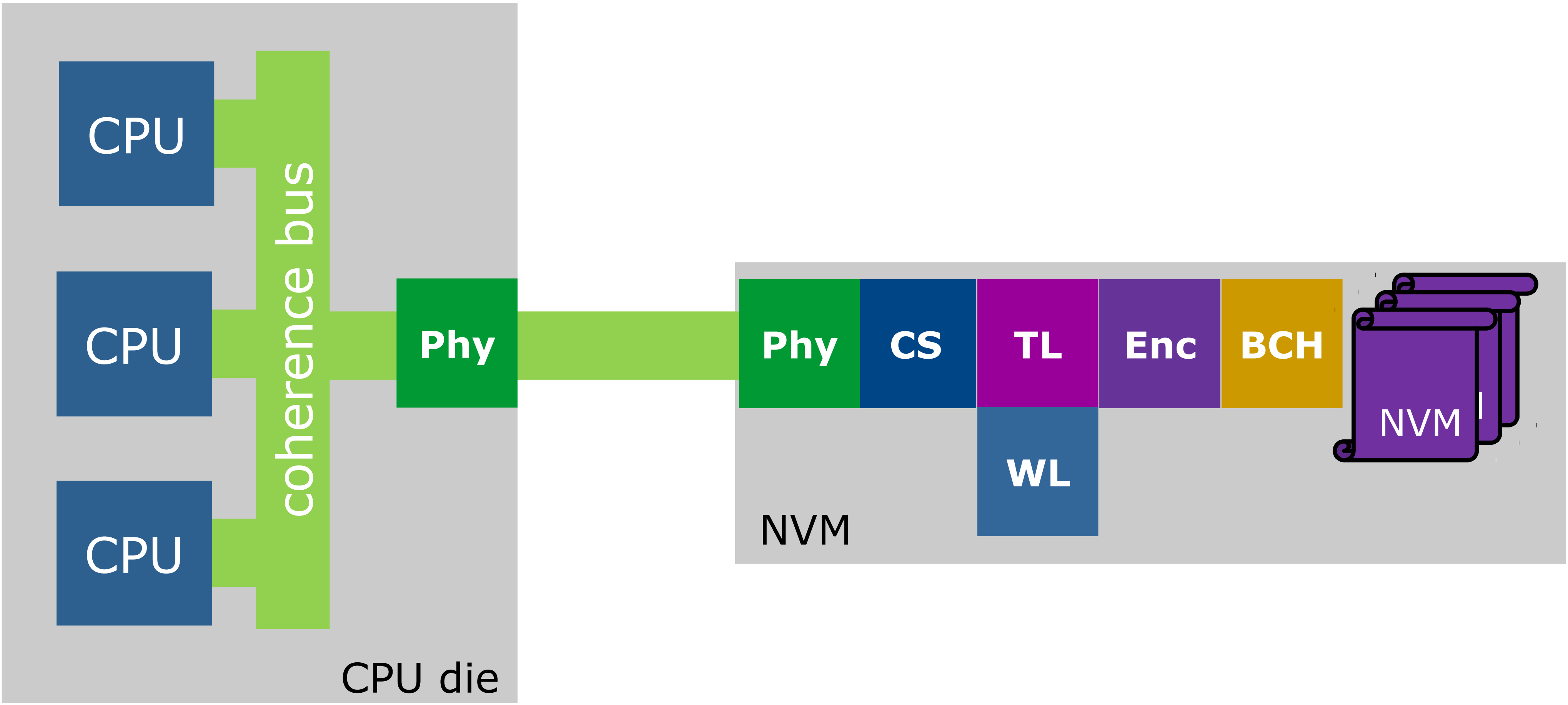
DRAM controller keeps coherence state



Don't do this


CPU
CPU
CPU
coherence bus
CS
TL
Enc
BCH
Log
Phy
WL
DDR
NVM
CPU die

Let's do this instead!

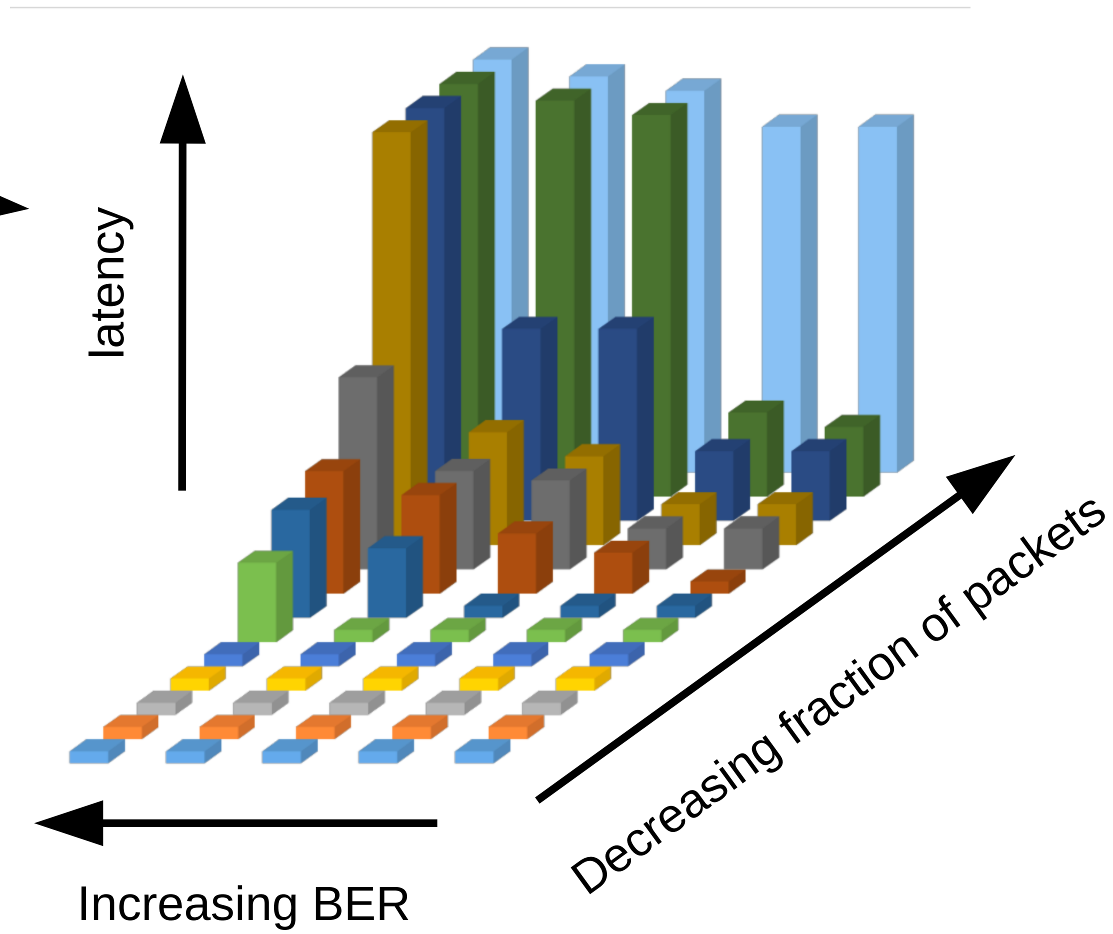


Why?

How?

Response latency jitter

- ECC: Hamming, BCH, LDPC?
  - BER too high for Hamming
  - LDPC not needed/too slow
  - BCH is well suited but variable latency
  - Code should be chosen for a particular NVM, don't try and put a universal engine on the CPU
- Other causes of variability in response times
  - Write/read asymmetry delays reads
  - Macroevents: overheating, wear leveling
- It is not cost-effective to architect resistive NVMs with deterministic latency
  - DDR/DIMM was not designed for jittery memory; coherence protocol was!



Wear leveling, data protection at rest

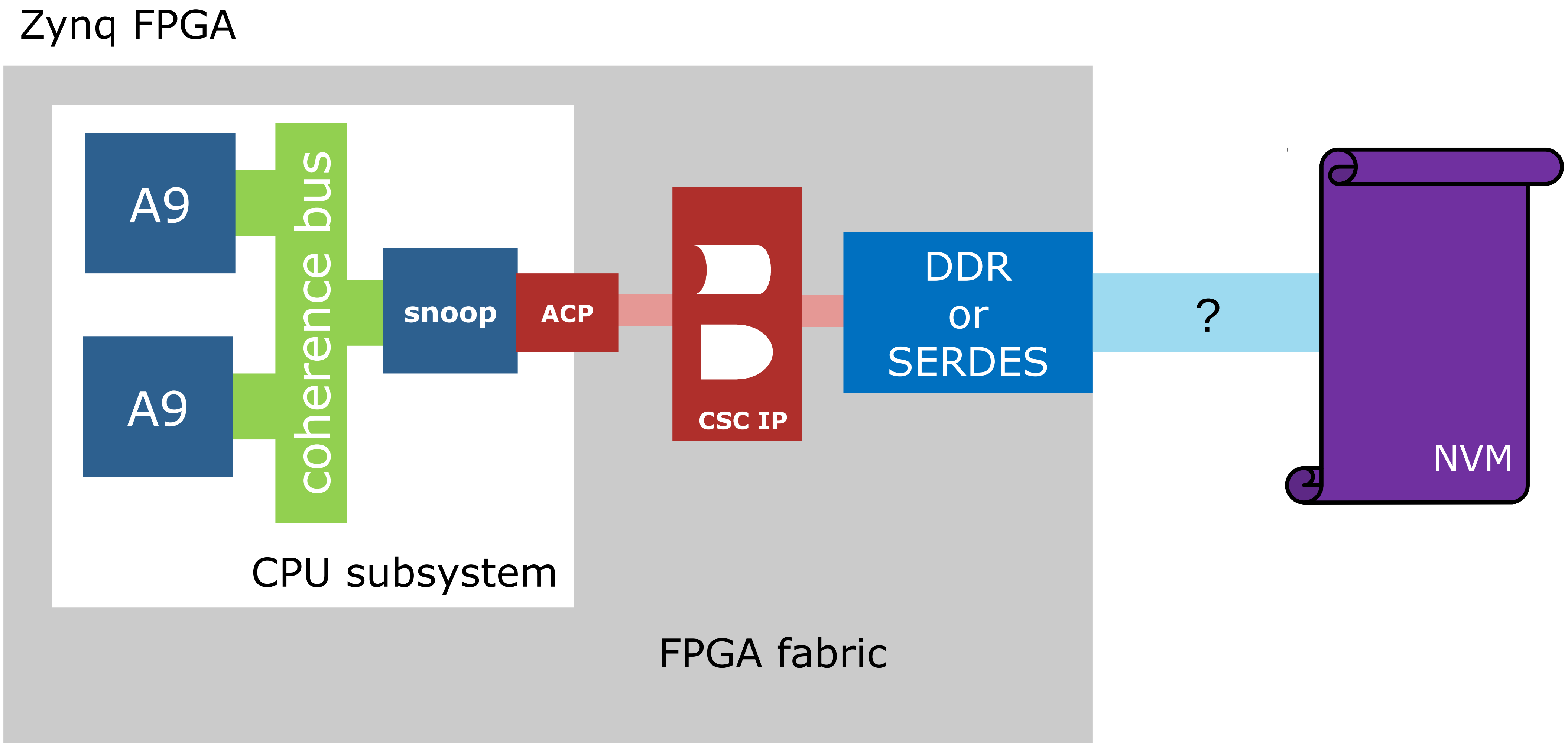
- Flash-like translation layer is too heavy, probably not needed
  - e.g. GB table for TB of memory
- Start-gap schemes are lightweight, but vulnerable to malicious code
  - May not be adequate for some types of resistive NVMs
  - Are you sure your scheme has no vulnerabilities?
- Aging controller
  - We have devised (and patented) translation schemes that are very fast, but have high up-front computational cost
  - One cost effective solution is to store pre-computed vectors as fuses in the controller
- Encryption of non-volatile working set
  - Scrubbing is not adequate, don't trust the programmer; interaction with wear leveling
- Hot-pluggable?

Controller belongs with non-volatile media!

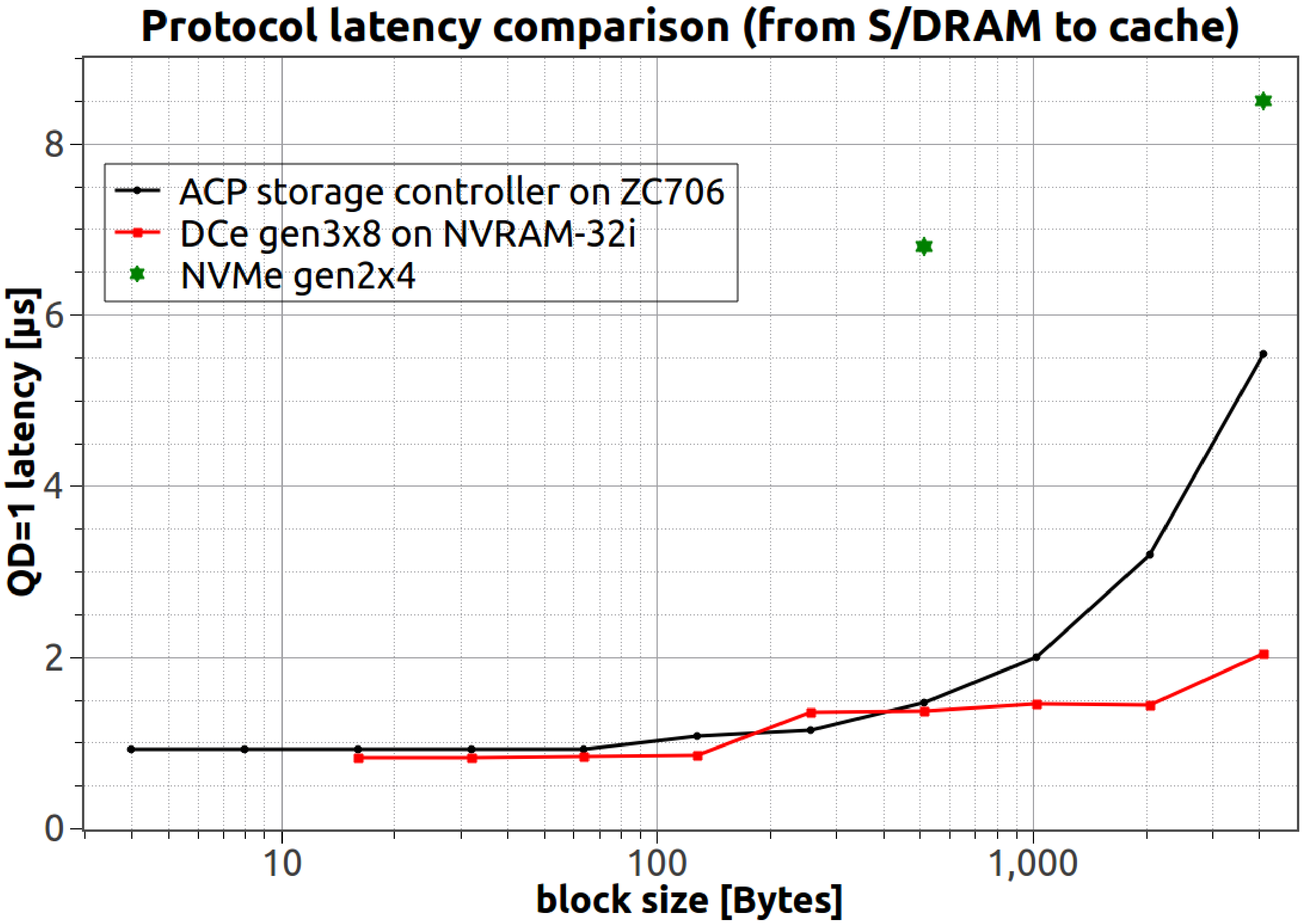
Why?

How?

Coherent storage controller in reconfigurable logic



Coherent storage controller in reconfigurable logic



- Comparable latency, but
  - CPU is 5-8x slower
  - Coherence bus is 12x slower
  - Cost is 2-50x lower

Coherent scale-out through external fabric

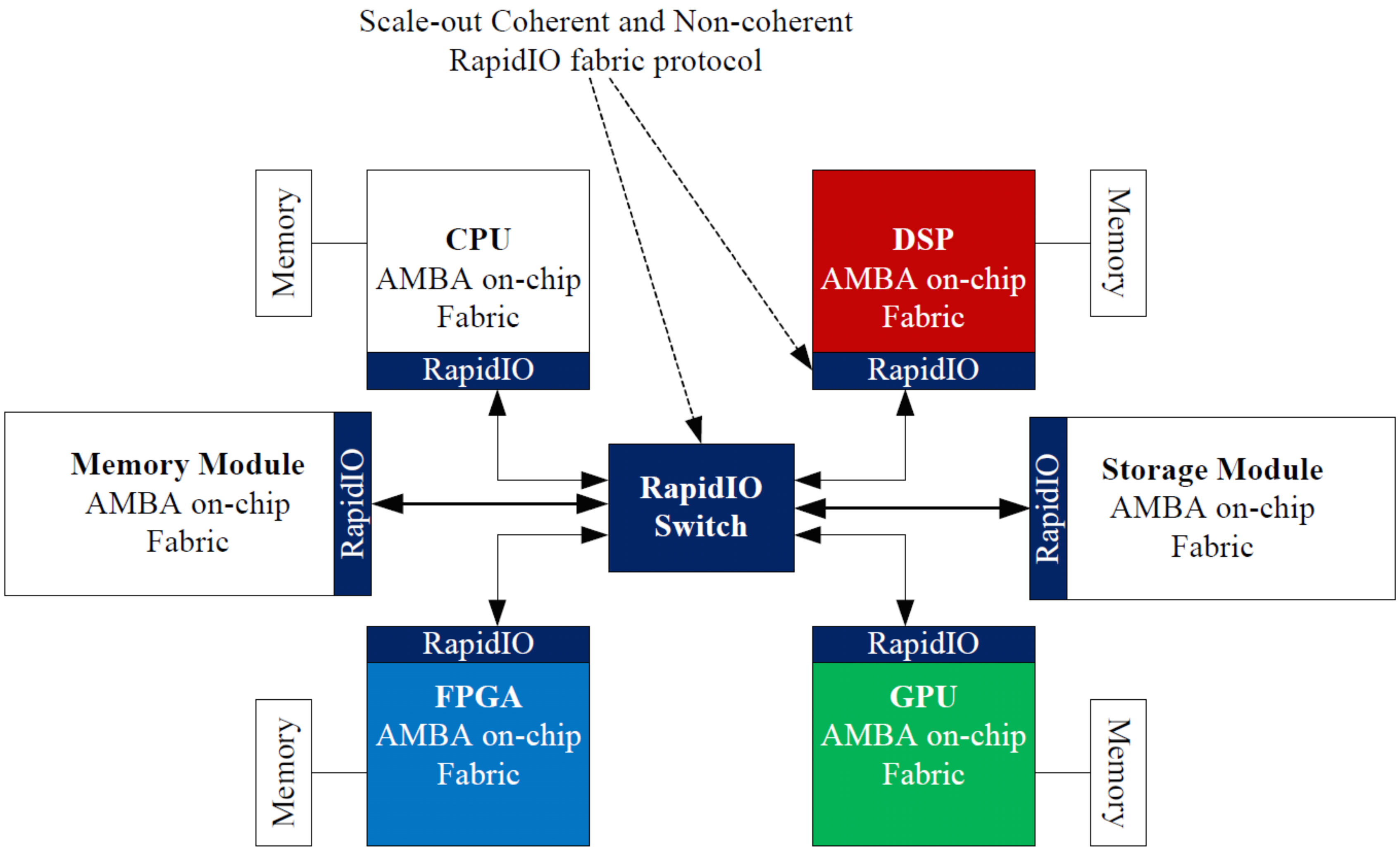


Figure 2-1. Heterogeneous Disaggregated Scale-out System Model with AMBA and RapidIO

Risc-V Shopping List*

- Hardware Coherence: Yes, please!
  - e.g. 300 Gib/s 40 ns on die, chip to chip
- Fast, wide ports for peripherals to join the coherence domain
  - e.g. opening into programmable logic, or scalable fabric (RapidIO?)
  - Unique advantage of the Risc-V ecosystem over competition
- Relinquish the non-volatile memory controller for now
  - Competing technologies make attempts at universal solution risky over the next decade
- Get used to high variability in main memory response time
  - Hyperthreads are bad memories' best friend

* for the enterprise

WD Western Digital®