CXL Native Memory™

Do We Really Need DDR?

Bill Gervasi, Principal Systems Architect

Wolley Inc.

bilge@wolleytech.com

the Future of Memory and Storage

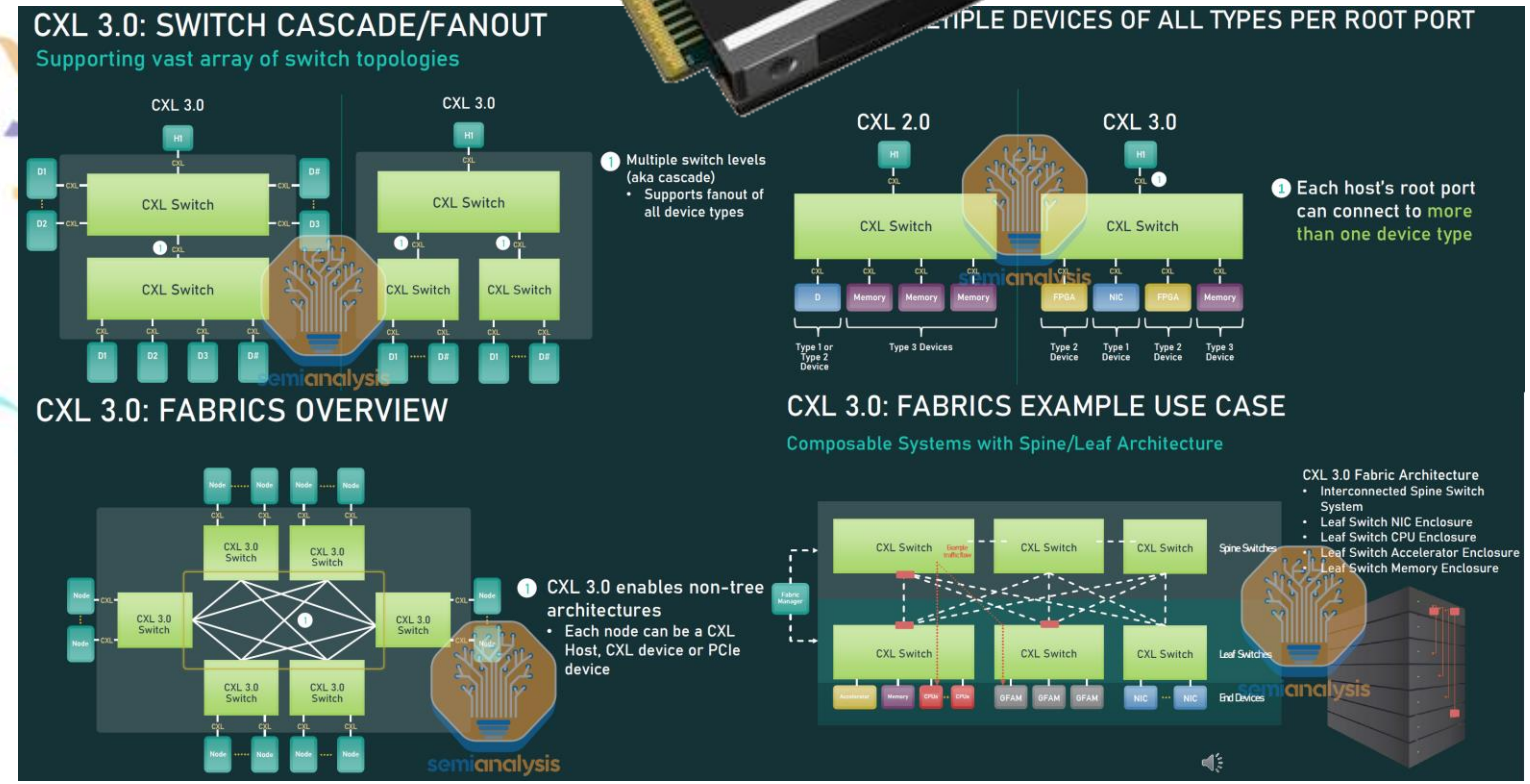
1. **CXL Native Memory: Do We Really Need DDR? (15 minutes)**

   CXL memory modules enable memory expansion, enabling larger capacities to support emerging applications such as large language models where the LLMs demand 140GB or more of local capacity. HBM can't enable these large memory capacities, and CXL is a logical method to expand memory, but at significant cost in terms of power consumed and bandwidth wasted. Is DDR doing us any favors, and can we imagine a memory world without DDR? CXL Native Memory proposes to replace the inefficient DDR interface with a CXL direct physical interface that drives memory cores from the CXL FLIT without protocol retranslation. CXL Native Memory reduces memory latency overhead while saving power.

Good news!

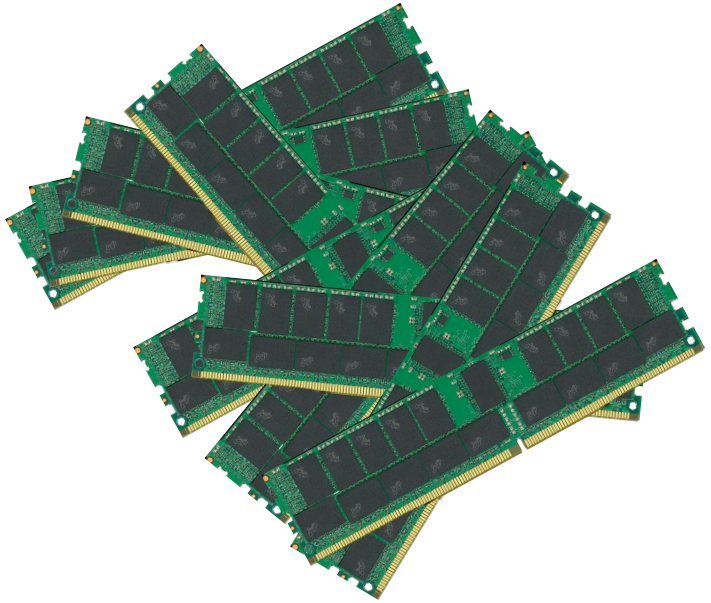
CXL ended the fabric wars

(sort of)



We'll address the impact of NVLink/UALink in my other talk…

Just as DDR5 goes to one DIMM per channel…



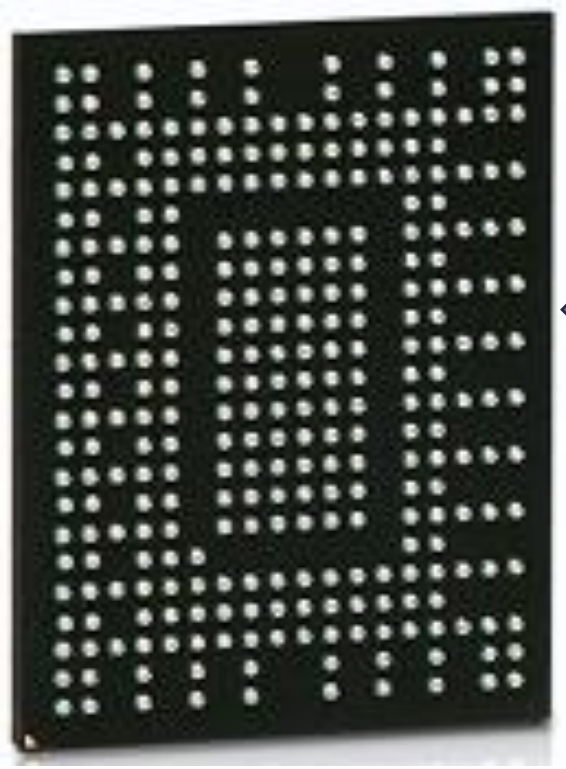
CXL comes along to save the day with DRAM expansion!

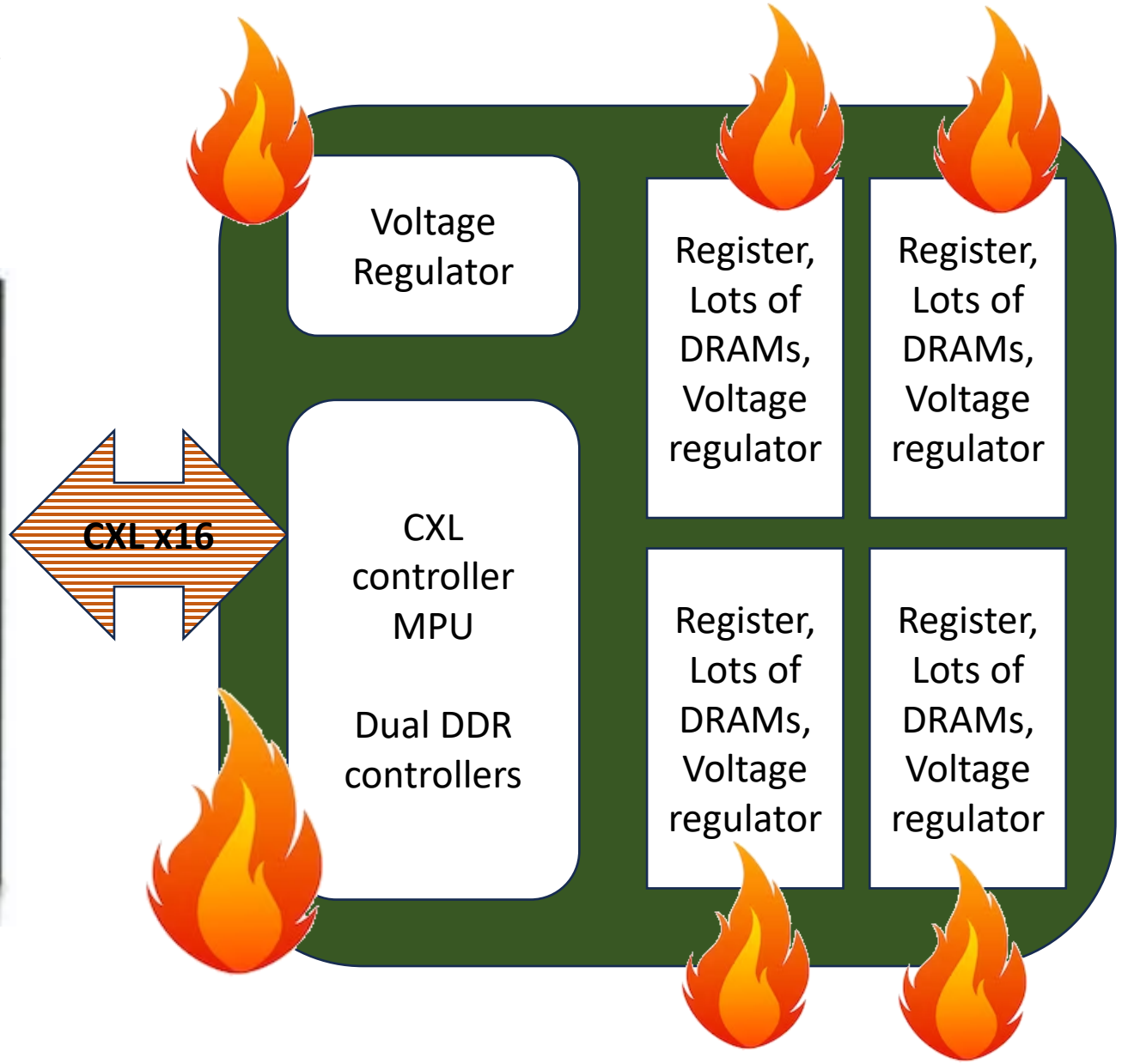
…but data centers are stuck with all these old DDR4 and DDR5 DIMMs they already paid for…

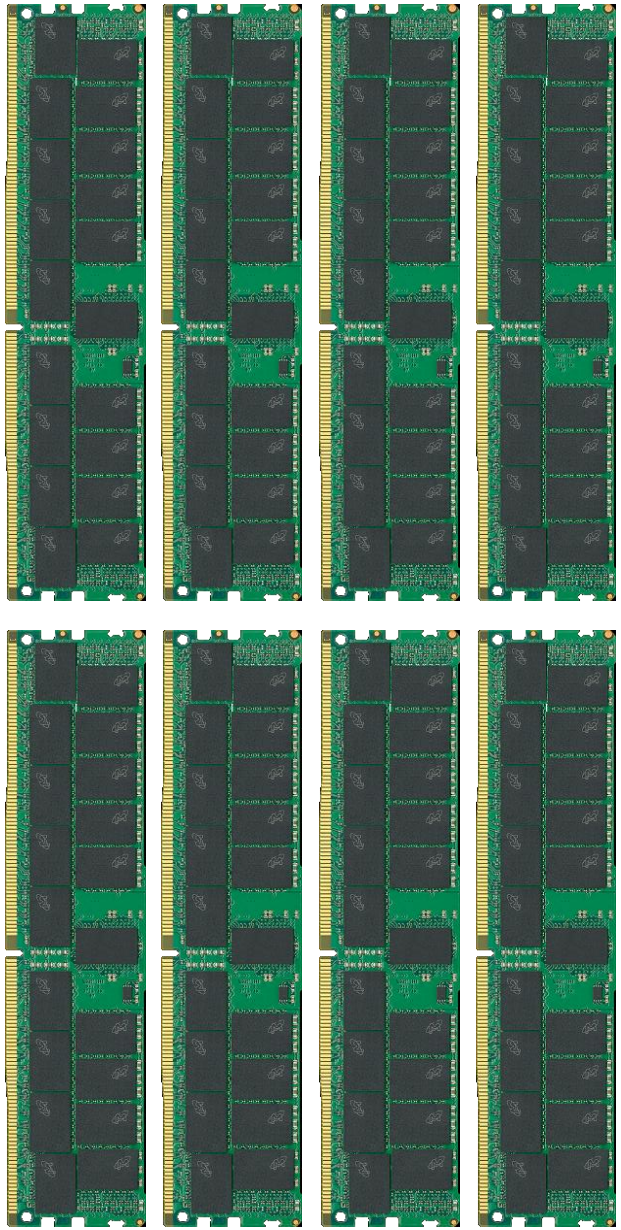
…cutting server memory capacity in half…

…so the initial introduction of an otherwise awesome technology is in the form of a chimera…

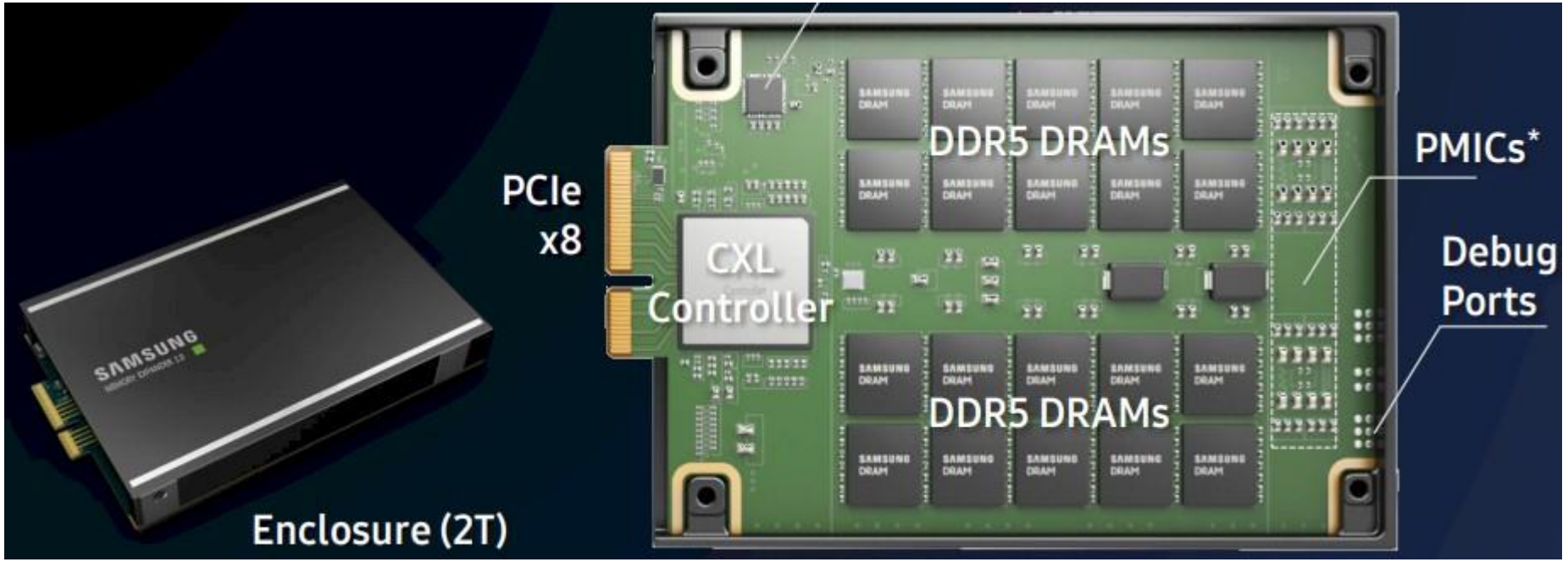
No wonder they are budgeting 75+W per card!







Fortunately, once the DIMM inventory is exhausted, the REAL CXL memory modules will take over…

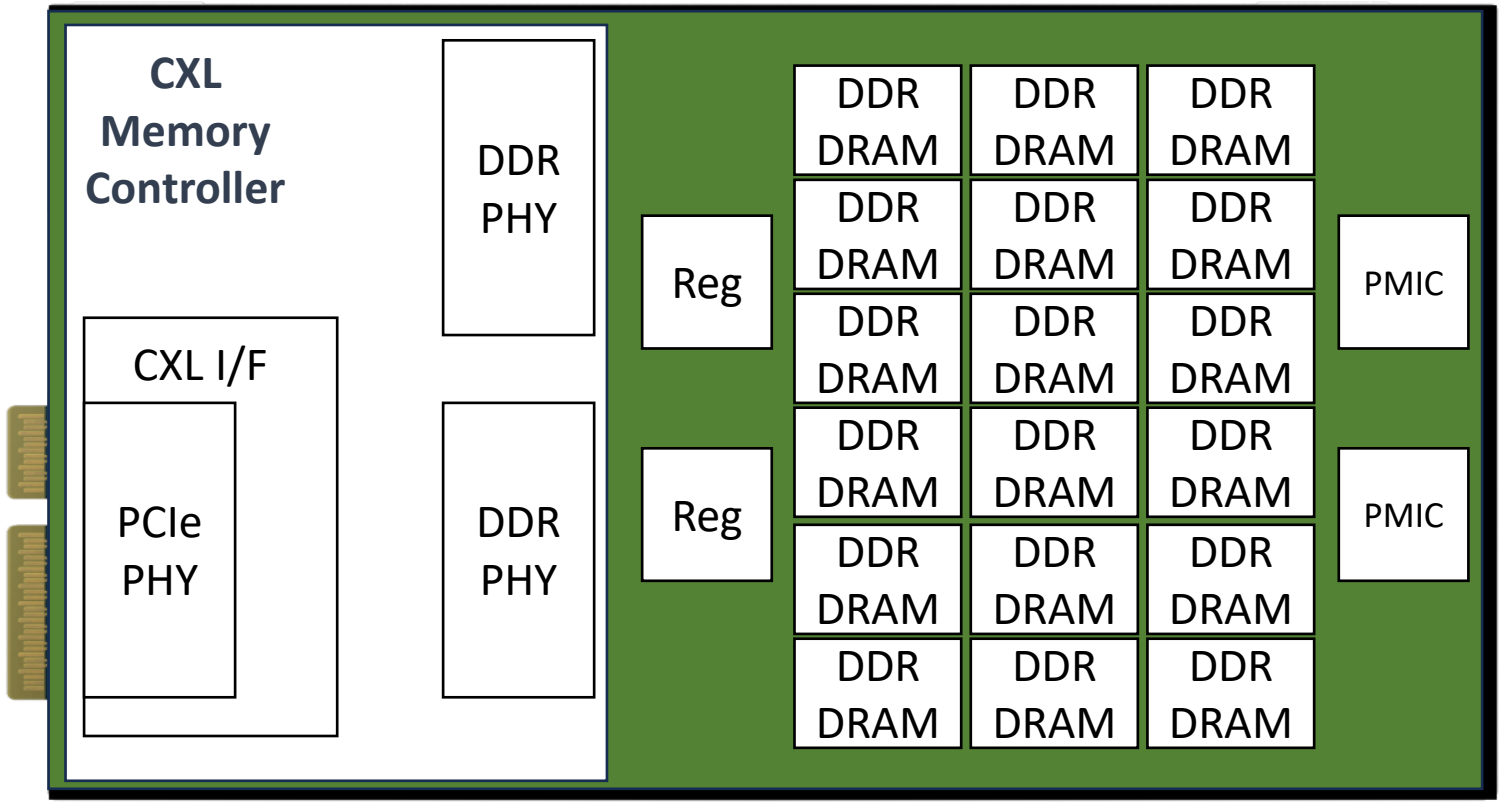


Eliminating redundant voltage regulation, sockets, etc…

More cost effective than a module populated with new DIMMs

Let's dig down another layer into the anatomy of a CXL module…

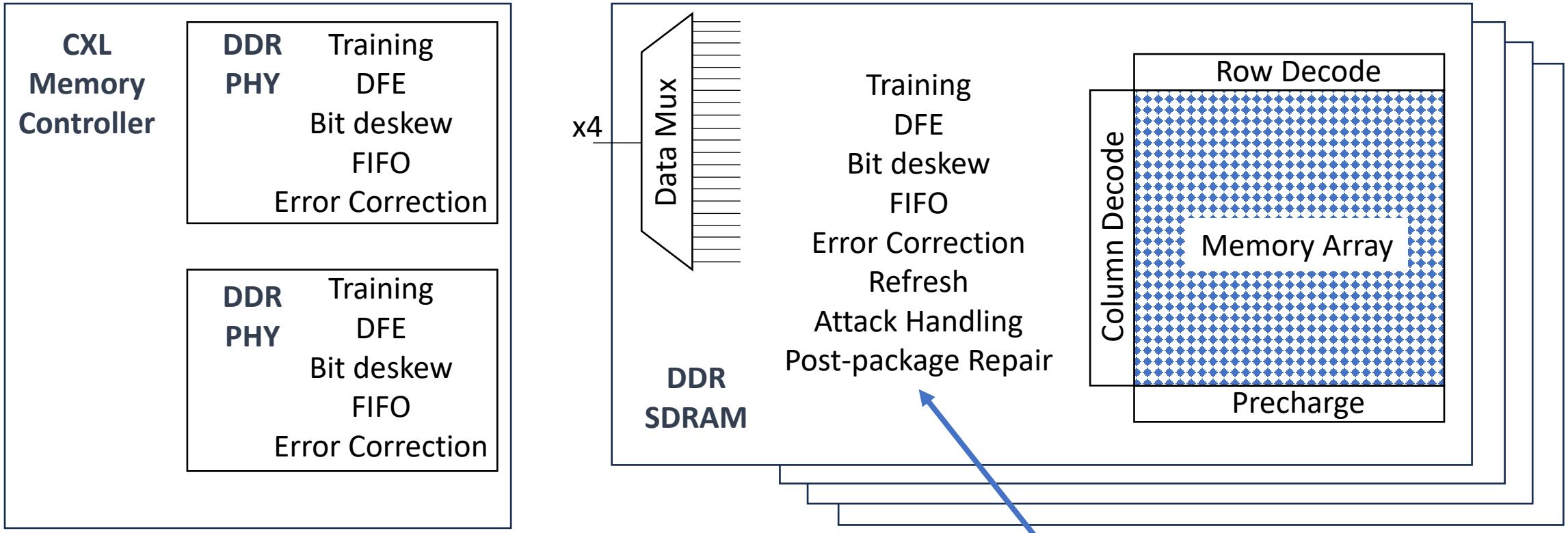
Long wires



CXL modules assume a very long PCIe bus requiring high current drivers

Each DDR PHY drives external circuits with heavy loading and complex calibration

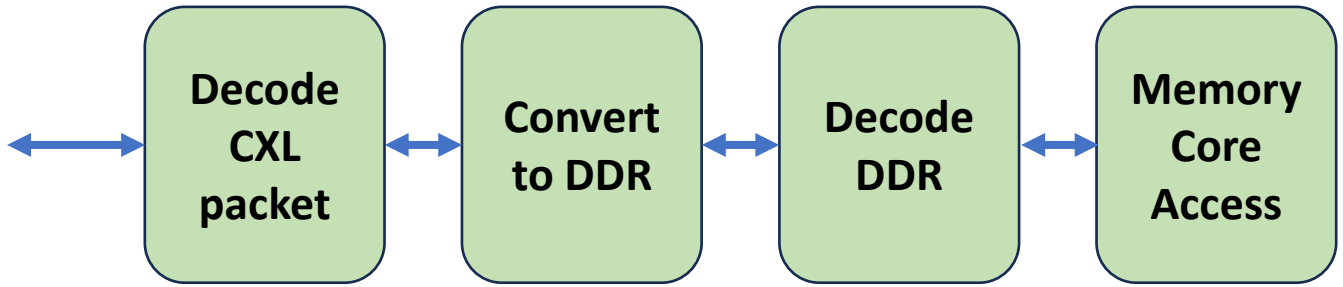
Redundant voltage regulation burns additional power



Drilling down one more layer, we see the redundancy in circuit design

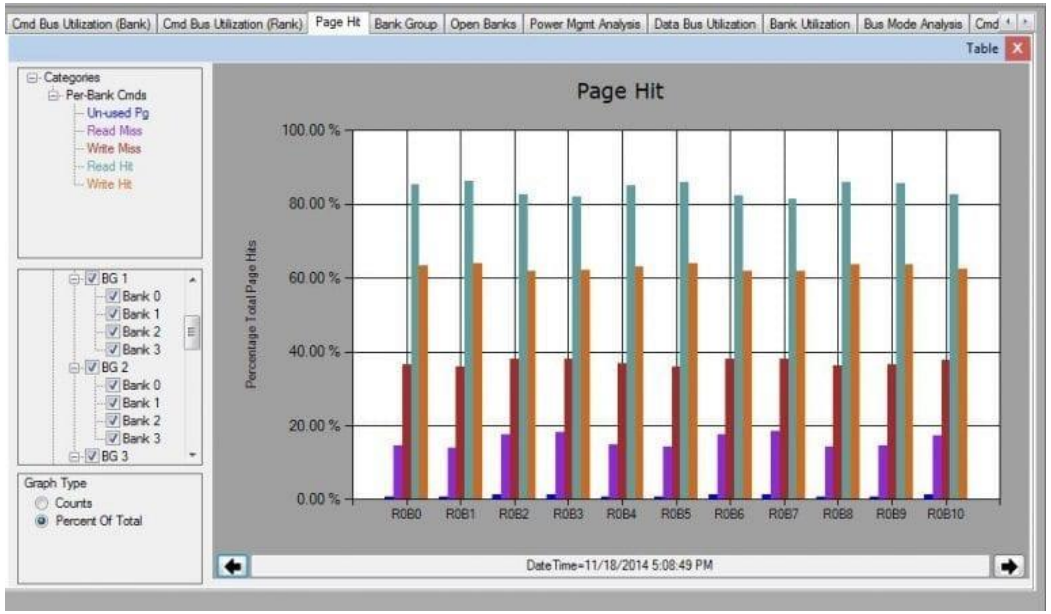
Duplicated in every DRAM
Paid for 80X over for one module
Burns power in every device

And we see the power and latency adders



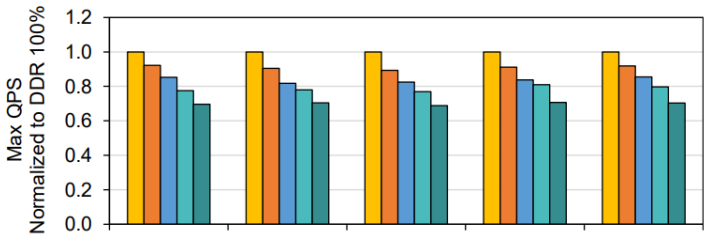
L1: 96% hit rate, 1 cycle access
L2: 95% hit rate, 25 cycles access
L3: 98% hit rate, 80 cycles access

The good news: near-CPU caches do have high hit rates (reduces waste from unnecessary accesses)



By the time an access gets to the local DRAM, though, hit rates start to drop dramatically

Read hit ~82%
Write hit ~62%



Access to remote memory drops even further, especially with increased thread count

Hit rate ~65%

…and this is before memory pooling…

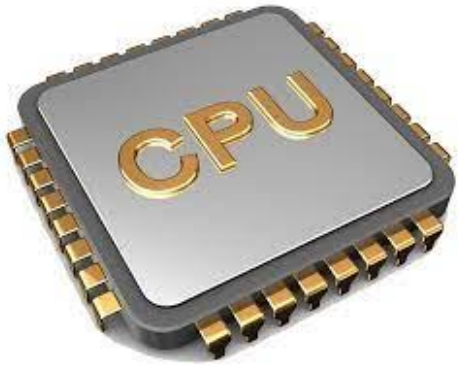
A question I have posed that CPU guys refuse to answer:

How much performance gain are we getting for each watt expended?

ESPECIALLY when it comes to speculative DRAM page

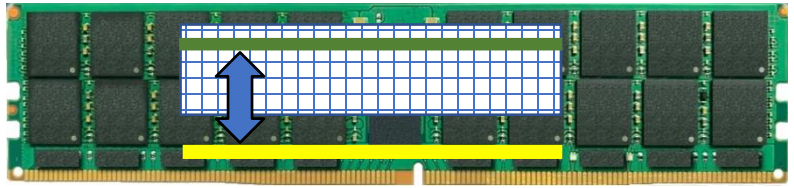
https://www.futureplus.com/blog/critical-memory-performance-metrics-for-ddr4-systems-page-hit-analysis

https://arxiv.org/pdf/2303.15375#:~:text=Meanwhile%2C%20as%20the%20block%20size%20increases%20beyond,latency%20begins%20to%20dominate%20the%20p99%20latency.



64 byte
cache line

1KB block X 10 DRAMs X 2 (ACT + PRE)



Waste > 99.97%

100 bytes used
on average

4KB block (plus DRAM accesses at SSD and Host)

Waste > 97.5%

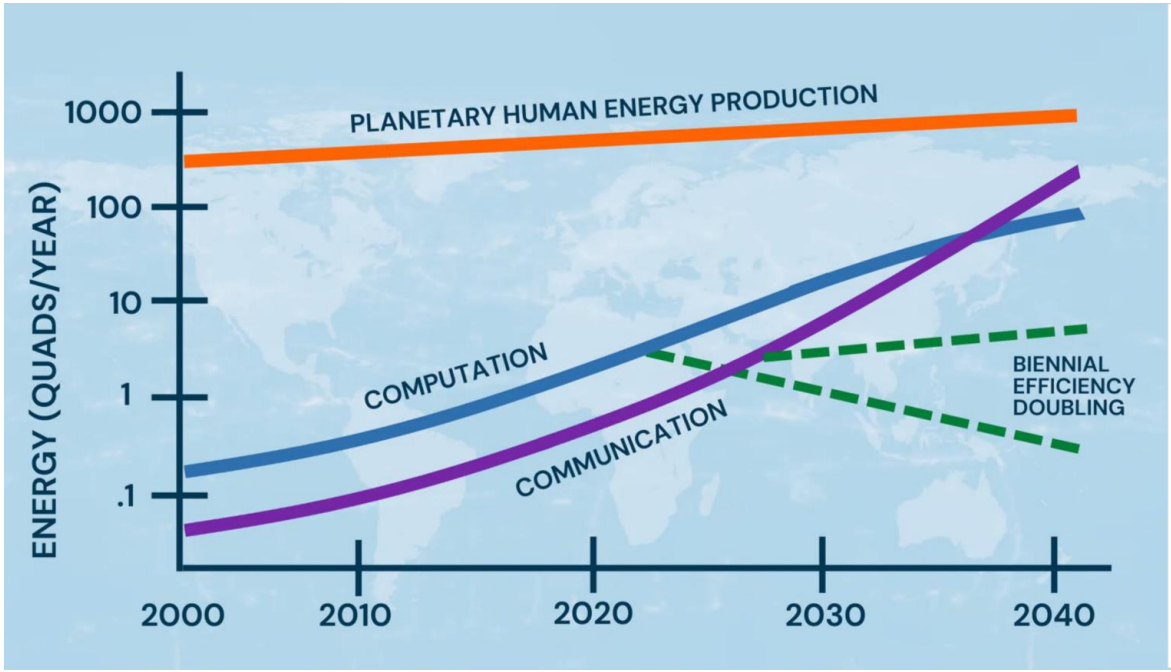
Adding up the ratio of data used to data moved, we can generously estimate that data centers are

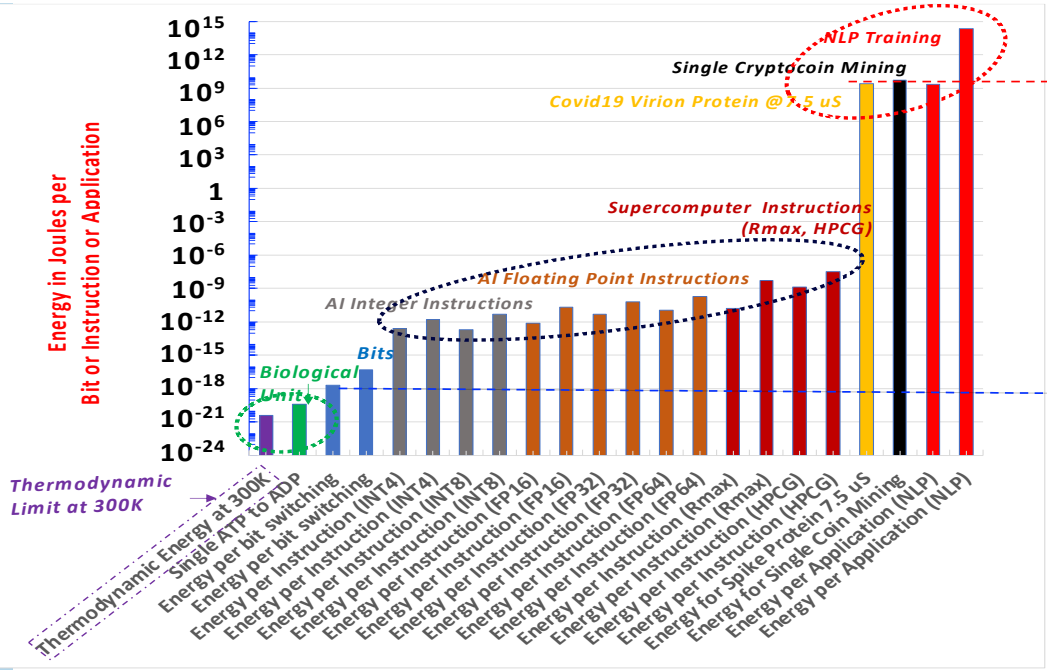
0.00004% efficient

(We suck at using data)

Who cares about data usage efficiency?

For starters, the US Department of Energy cares about avoiding a time when we can no longer power the internet





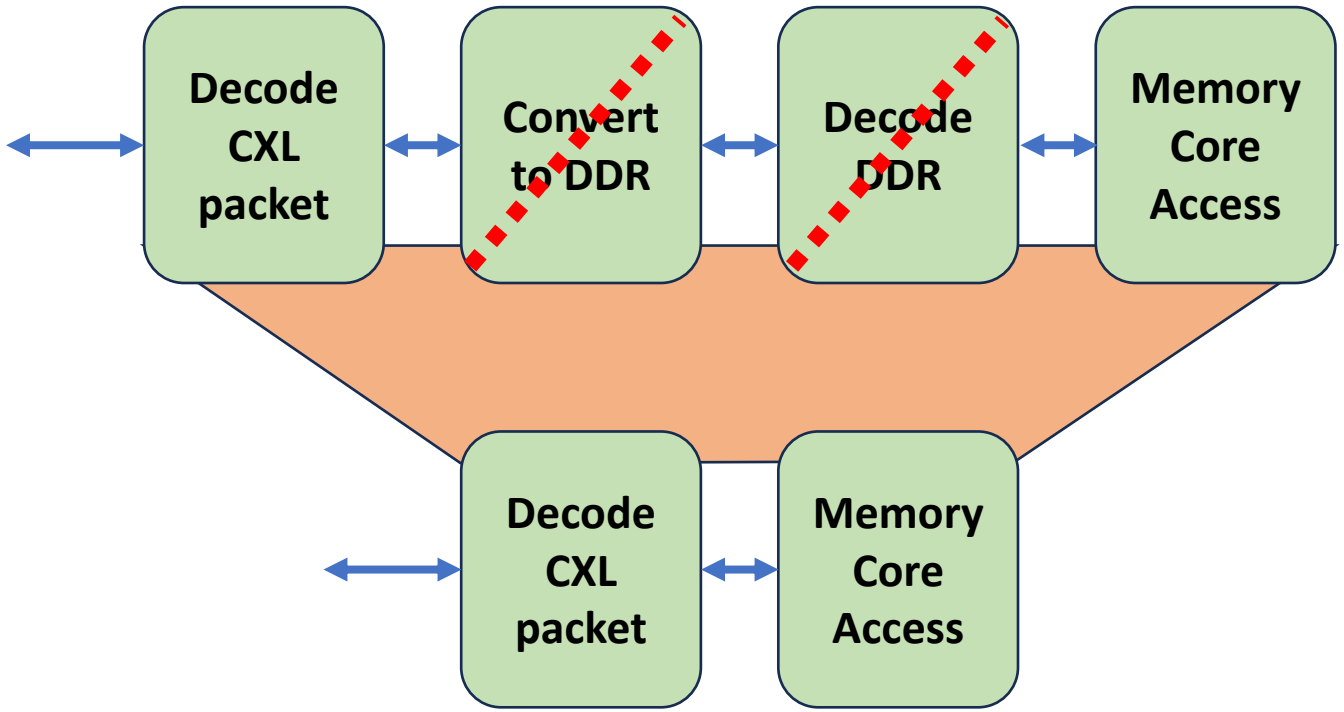
10^{9} to 10^{15}

10^{-18} to 10^{-21}

27 to 36 orders of magnitude
(From Algorithms to Atoms)

Fortunately, large data center owners are finally catching on to the idea that total cost of ownership matters

CXL Native Memory™ Imagines a World Without DDR

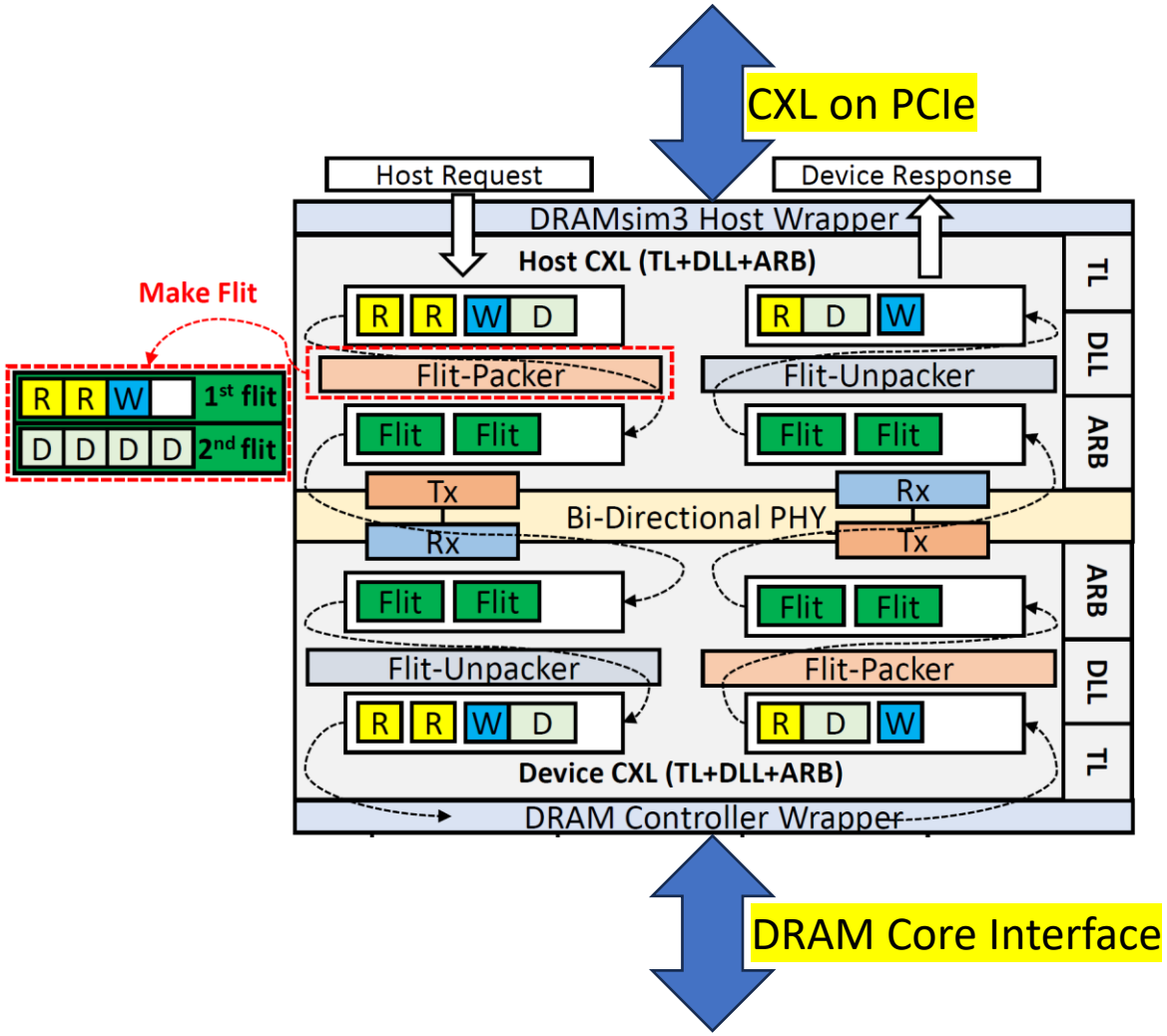


Current CXL DDR Memory Architecture

CXL Native Memory Architecture

And we see the power and latency improvement

CXL Native Memory Uses the CXL FLIT Directly



CXL FLIT has everything a memory needs

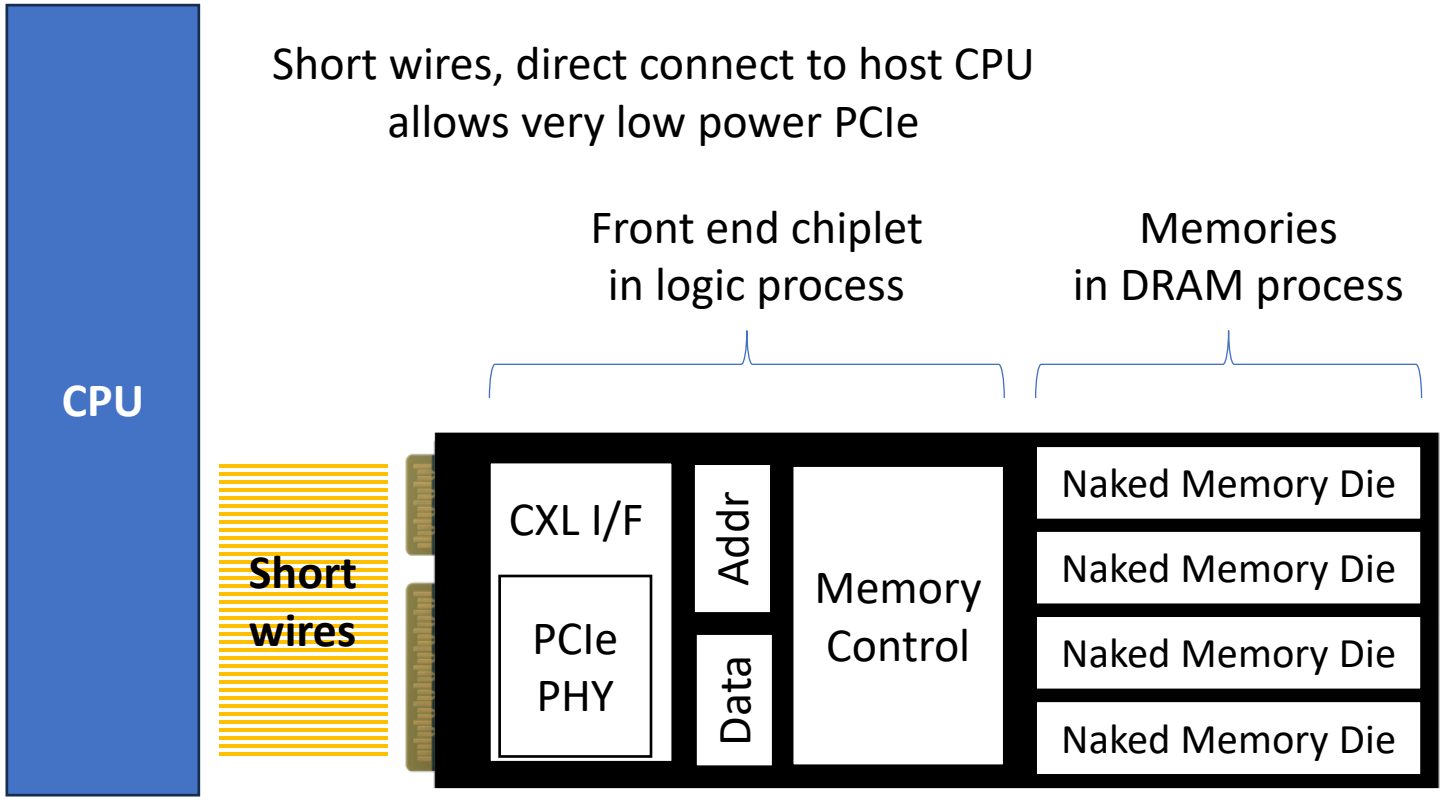
- Address
- Command
- Data + metadata

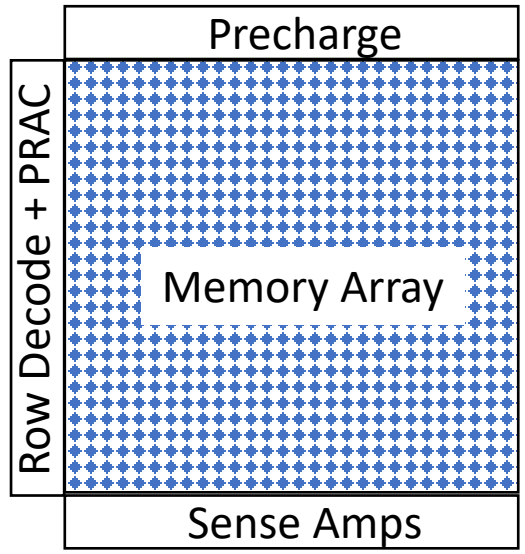
Translate to core functions and timing (banks, rows, columns, etc.)

No DDR interface is needed

CXL.io provides for interesting enhancements to strict memory protocol

Bringing CXL to the Motherboard



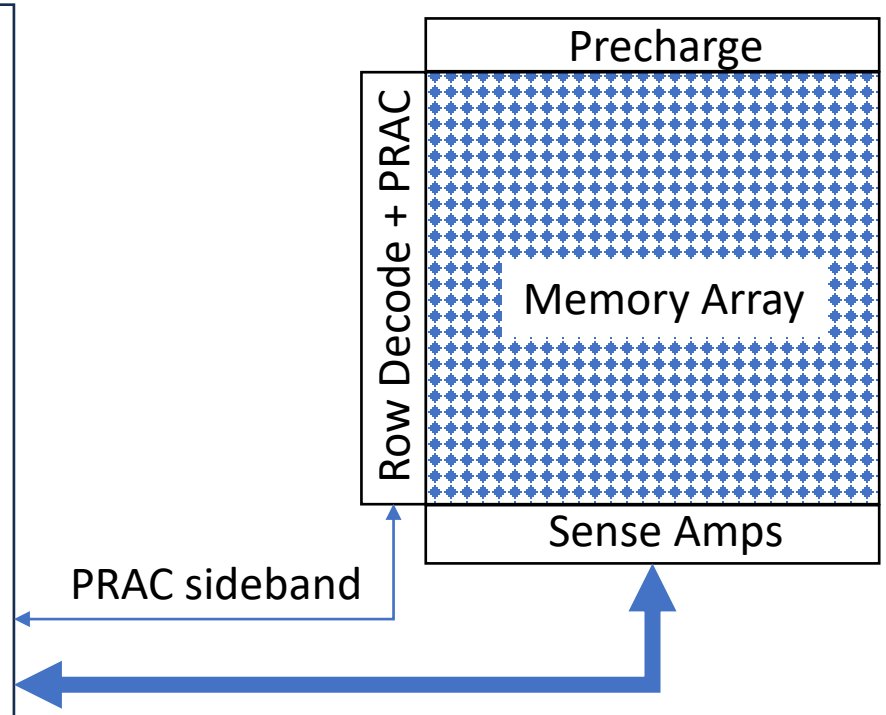


Naked Memory die are just memory arrays, row drivers, and sense amps

CXL Memory Controller

Bit deskew
FIFO
Error Correction
Refresh
Attack Handling
Post-package Repair

Redundancy eliminated



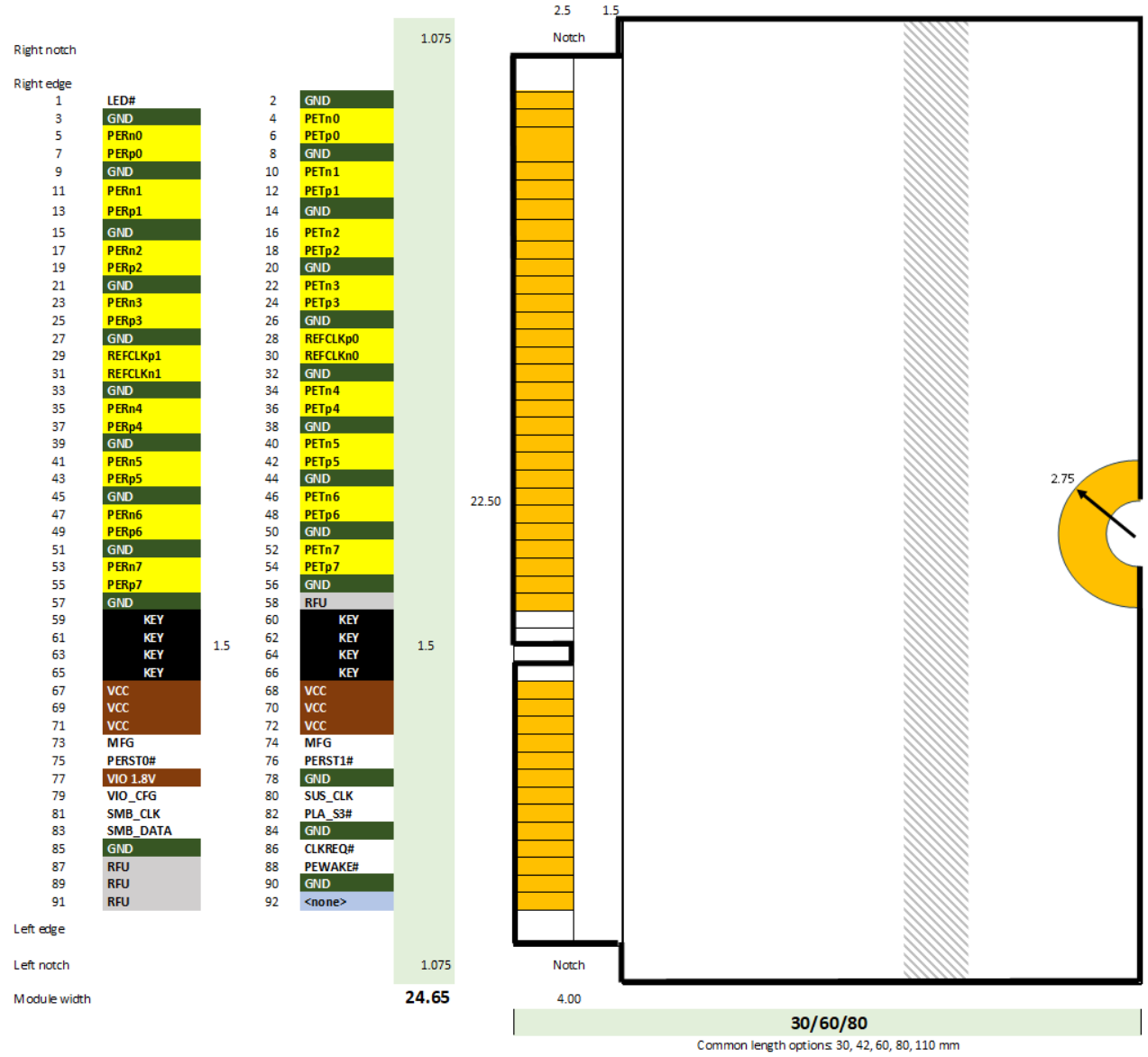
Control logic is not duplicated in every RAM but is consolidated in a single controller

Wide I/O interface allows for relaxed memory core timing

Wide I/O up to 512 bits + ECC + Metadata

ONE SMALL FLIT IN EVERY TRANSFER
LARGE FLITS ARE MULTIPLES OF SMALL FLITS

Data Usage Efficiency = 2000X that of CXL-DDR



Proposal for FleX (M.28)

FleX (M.28) has

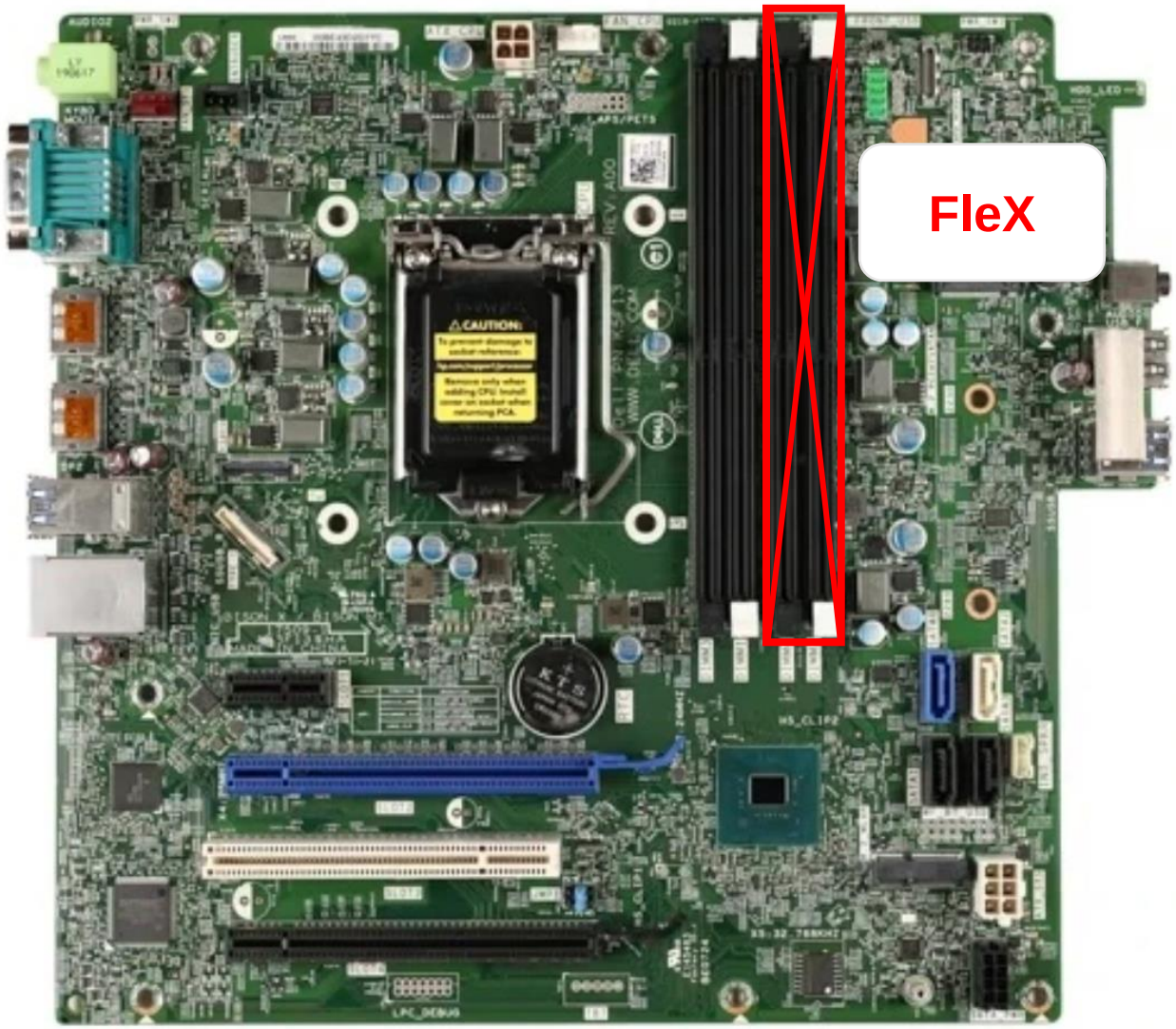
- PCIe Gen 6 x8 support + CXL
- Diff pairs on same side of module for Gen6 support
  - Tx, Rx calibrated independently
- On-module regulation
- Power ~ 11W

M.4 lengths TBD; starting estimates:

- 30 mm
- 60 mm
- 80 mm

Actual ratios:

2530	2560	2580



DDR5 has a “slot problem”

To run DDR5 with two DIMMs per channel, the channel maxes out at 5600

To run DDR5 at 6400+, the layout is restricted to one DIMM per channel

This means end users must choose between speed and capacity

Sad user

CXL Native Memory in a FleX module allows DDR slots to run at 6400+ without sacrificing memory capacity

Courtesy of Tom Schnell, Distinguished Scientist, Dell Computer

Summary

CXL allows for memory expansion

CXL memory allows DDR to run at full speed

CXL memory modules are not power efficient

DDR5 is hitting a capacity wall

DDR is not needed for CXL

Great for applications that need gobs of memory

CXL Native Memory drives array from FLIT

Low pincount way to expand memory

FleX module brings CXL to motherboards

Thank you for your time

Any more questions?

Bill Gervasi, Principal Systems Architect
Wolley Inc.
bilge@wolleytech.com