

**IBM Power10 & z16:**

**Extreme OMI Exploitation  
For Main tier Memory**

Bill Starke

August 1, 2022

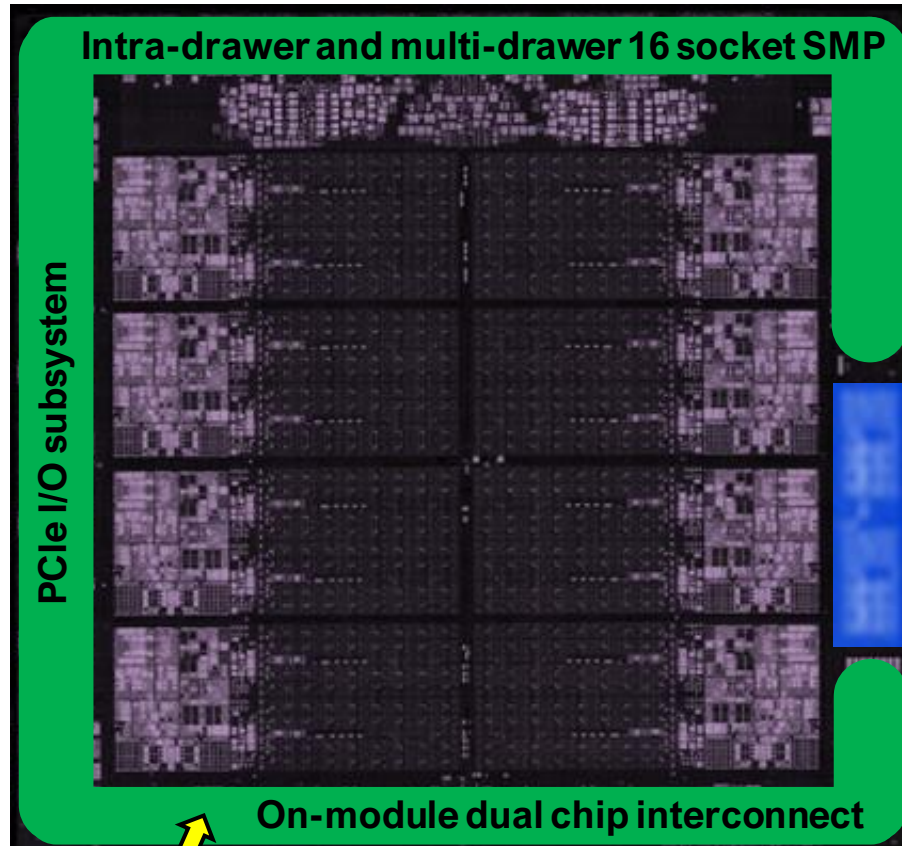


# Memory Taxonomy (OMI and CXL.mem are Complementary)

<u>Memory Type</u>	<u>Typical Industry Solution</u>	<u>OMI Strategy</u>
Near Tier (Bandwidth)	HBM on-module memory stacks (limited capacity, custom processor, custom system)	OMI attached GDDR/LPDDR D-DIMM (approaching HBM bandwidth)
Main Tier (General Purpose, <b>Latency</b> )	DDR direct attach DRAM DIMM (good latency, limited bandwidth)	OMI attached DRAM D-DIMM (good RAS, bandwidth, capacity, latency)
Private Far Tier (Cost, Capacity)	CXL.mem attached DRAM-derivative CXL.mem attached Storage class	(leverage industry CXL) CXL.mem attached DRAM-derivative CXL.mem attached Storage class
Shared Far Tier (Cost, Capacity, Flexibility)	CXL.mem attached DRAM-derivative CXL.mem attached Storage class	(leverage industry CXL) CXL.mem attached DRAM-derivative CXL.mem attached Storage class
Clustered (share all tiers) (Flexibility, Capability)	Future? CXL multi-host cluster	Memory Inception (Power architecture)

# z16 Mainframe OMI: Reliability, Capacity & Bandwidth Density

IBM Telum Processor Chip  
(used in z16 mainframe system)



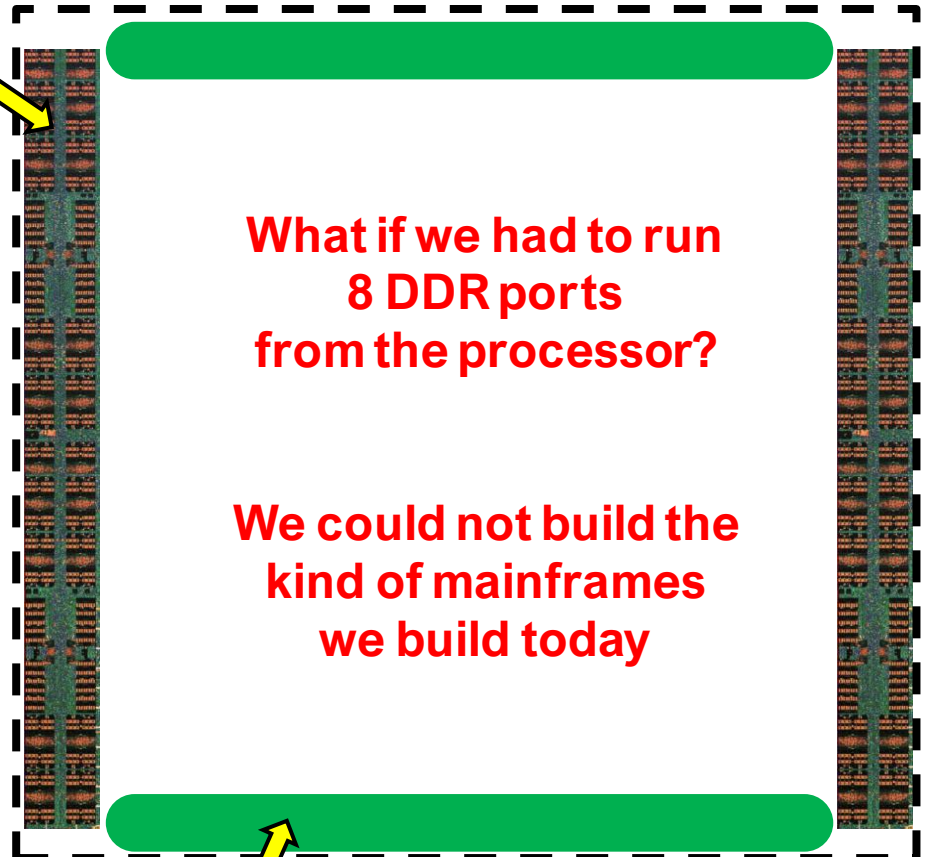
Substantial PHY beachfront available for other critical high bandwidth interfaces

OMI PHYs  
(x4/x4 attach to 8 buffers)

Fan-out to 8 DIMMs

Robust packet Based transport

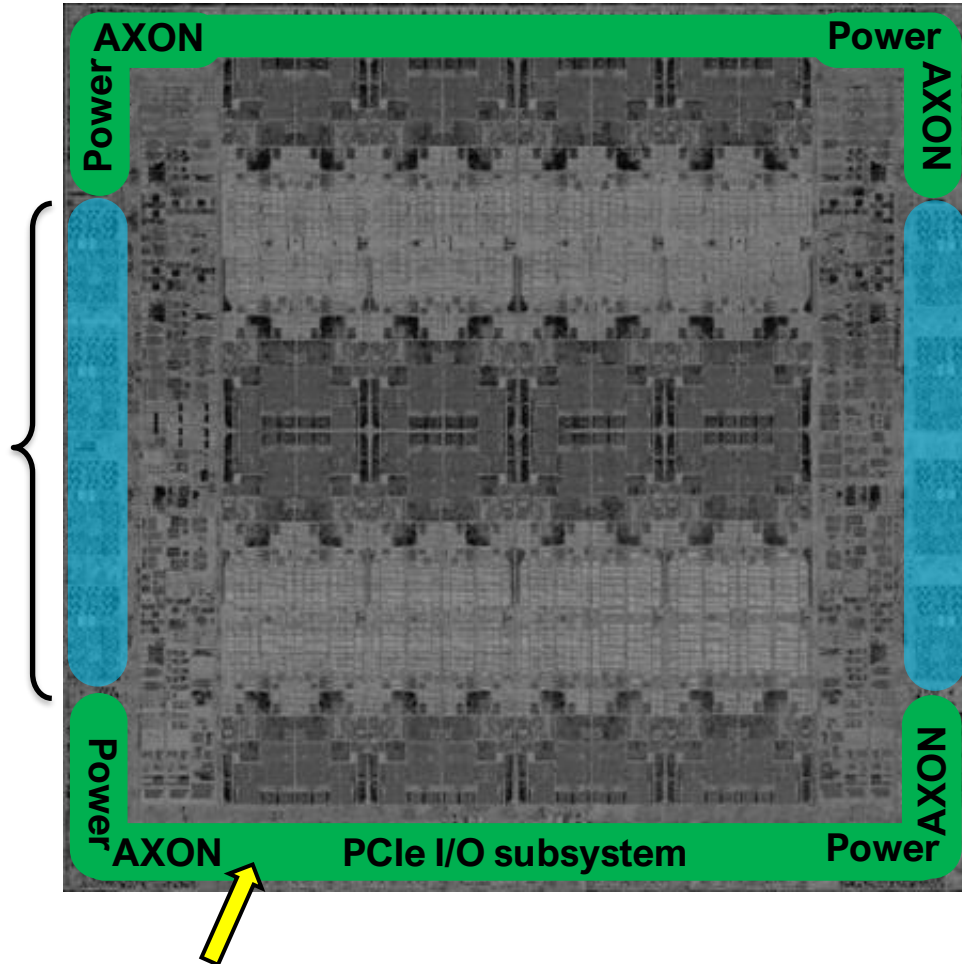
Where would we be without buffered memory?



Massive loss of PHY beachfront for other critical high bandwidth interfaces

# Power10 OMI: Reliability, Bandwidth, Capacity

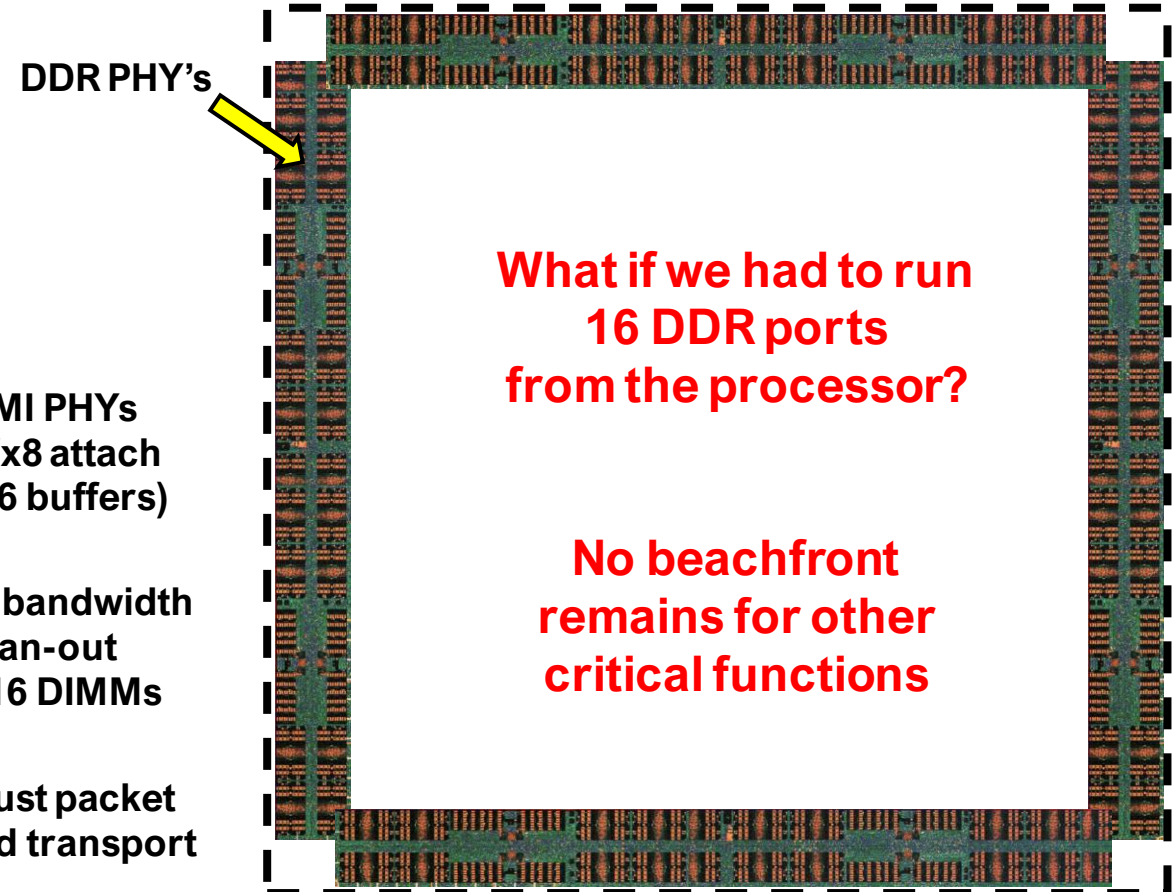
## IBM Power10 Processor Chip



Substantial PHY beachfront available for other critical high bandwidth interfaces

- OMI PHYs (x8/x8 attach to 16 buffers)
- High bandwidth Fan-out to 16 DIMMs
- Robust packet Based transport

Where would we be without buffered memory?



Complete loss of PHY beachfront for other critical high bandwidth interfaces

**You ask:**

**So that's fantastic,  
but can't I do something similar with CXL.mem?**

**You ask:**

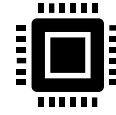
**So that's fantastic,  
but can't I do something similar with CXL.mem?**

**You should ask:**

**How can you get to a reasonable main tier latency  
with a buffered solution instead of DDR direct?**

# History of Buffered Main-tier Memory for IBM Z and Power

Late 1990's through mid-2010's → **Good Latency**



- DDR controller on processor
- High speed ECC protected interface running at N:1 multiple of DDR speed
- “Puppet-string” buffer (normally on DIMM) downshifts to DDR speed and drives DDR PHY
- Multiple DDR ports per buffer

**Reliability, Beachfront Density, Fan-out for Capacity and Bandwidth**

2014: Centaur Agnostic Buffer → **Better Latency**



- DDR controller, **cache, read-update path** on Buffer
- High level, **technology agnostic**, packetized read/write/data protocol from processor to buffer
- High-speed SERDES with **CRC / replay**
- Maintaining N:1 DDR multiple, speculation, and associated pipelining critical for **latency**

2019: OMI Buffer → **Best Latency (less than 10ns adder beyond DDR direct)**



- **Non-proprietary**, OpenCAPI command subset (plug-and-play objective)
- **Lightweight form factor**
- N:1 DDR multiple (maintain **latency optimizations**)

Meanwhile, in the Broader Industry:

**Buffered Memory gets a bad reputation due to Latency problems**



# The Other side of OMI's Lineage

## 2014: IBM POWER8 / POWER9 with CAPI



- Early coherent accelerator attach capability
- Has memory home agent function
- Built on top of PCIe transport (complex PCI function adds latency)
- **Fine for far tier memory latency**
- **Not even close to being adequate for main tier mem latency**

## 2017: IBM POWER9 with OpenCAPI



- Major focus on latency reduction
- Runs native link layer over ethernet-type PHY
- PCI overheads completely removed
- **Fine for far tier memory latency**
- **Much better for accelerators, but still not good enough for main tier memory latency**

## 2019: OMI



- Stripped down to OpenCAPI memory home agent function only
- Physically plumbed through latency-optimized processor memory controller unit
- Memory controller built from Centaur-based IP, N:1 clocking with DDR, fast paths, etc....
- **Substantial latency improvements: competitive for main tier or far tier usage**



# Where does CXL.mem fit in?

## 2014: IBM POWER8 / POWER9 with CAPI

- Early coherent accelerator attach capability
- Has memory home agent function
- Built on top of PCIe transport (complex PCI function adds latency)
- **Fine for far tier memory latency**
- **Not even close to being adequate for main tier mem latency**



## 2017: IBM POWER9 with OpenCAPI

- Major focus on latency reduction
- Runs native link layer over ethernet-type PHY
- PCI overheads completely removed
- **Fine for far tier memory latency**
- **Much better for accelerators, but still not good enough for main tier memory latency**



## 2019: OMI

- Stripped down to OpenCAPI memory home agent function only
- Physically plumbed through latency-optimized processor memory controller unit
- Memory controller built from Centaur-based IP, N:1 clocking with DDR, fast paths, etc....
- **Substantial latency improvements: competitive for main tier or far tier usage**

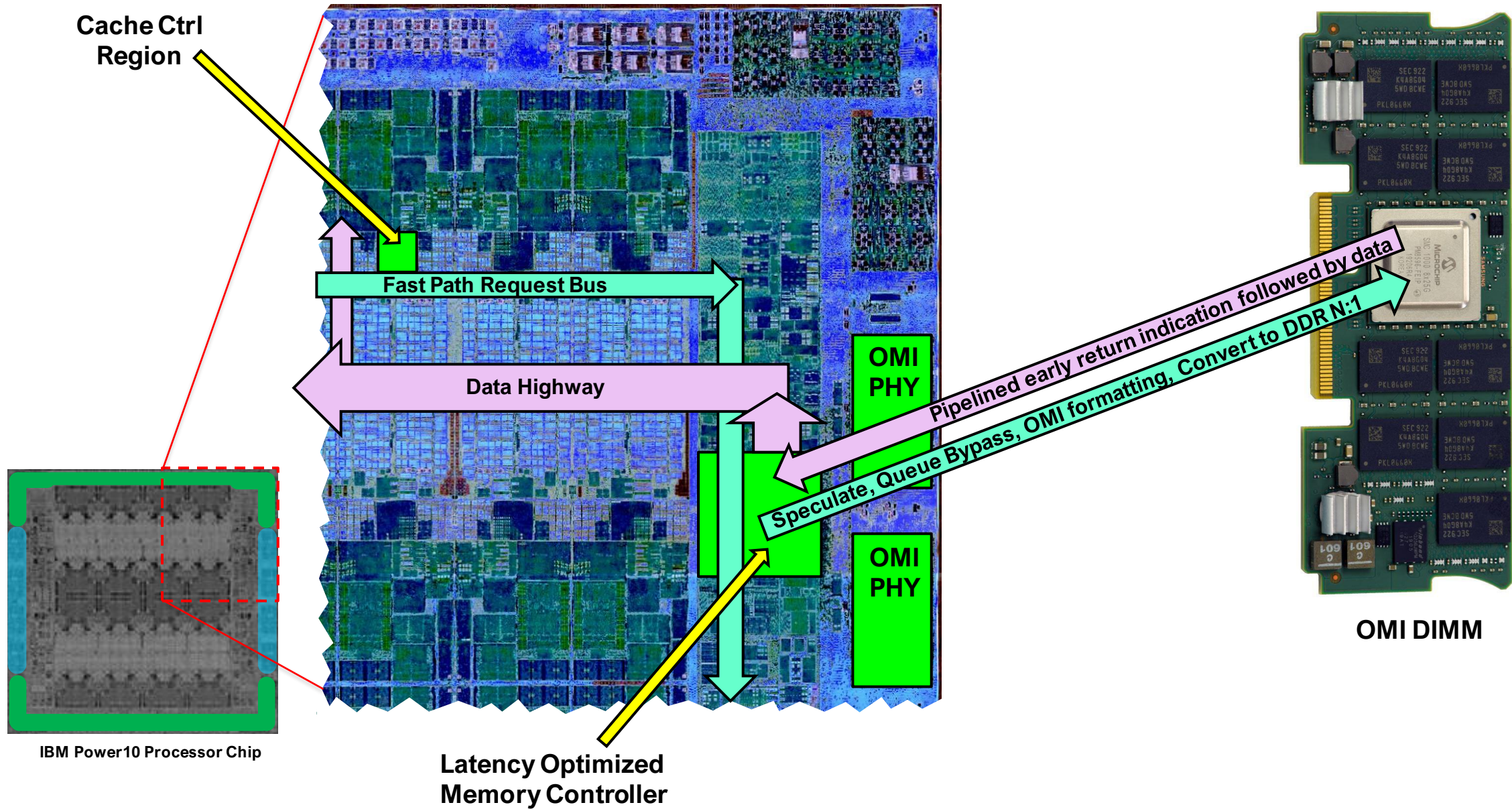


## Today: CXL

- Built on PCI (better than 2014)
- Great for acceleration
- Great for far tier mem
- But not trying to be a DDR replacement

You are Here

# Power10 OMI: Extreme Latency Optimization



## **The OMI Challenge:**

**DDR is running out of gas.  
CXL.mem is great for far tier scenarios.**

**If you think CXL.mem is fine for main tier memory,  
let me see you rip out all your DDR ports and replace  
them with CXL.mem ports.**

**IBM Power and System Z (mainframe) have no DDR ports.  
We use OMI instead for main tier memory.  
Maybe you should consider it.**



When you interact with IBM, this serves as your authorization to Flash Memory Summit or its vendor to provide your contact information to IBM in order for IBM to follow up on your interaction.

IBM's use of your contact information is governed by the IBM Privacy Policy.

