White Paper

CXL Native Memory re-imagines how modular memory expansion can be delivered

CXL Native Memory™ Enables Modular Solutions for Scaling the Memory Wall

By Bill Gervasi and San Chang

Abstract: Data centers hit the memory wall as increases in DRAM capacity slowed and the number of memory modules per channel decreased. Just in time, the fabric wars ended and CXL emerged as the primary alternative to the DDR memory modules that traditionally supplied main memory. Now that the infrastructure for CXL is expanding, Wolley introduces a variant for extending the application of CXL into embedded systems such as mobile applications, industrial systems, and artificial intelligence accelerators.

CXL Native memory combines the best features of CXL Memory Modules with innovative simplifications to offer higher performance at lower power than comparable solutions.

Introduction

The industry is in the process of adopting Compute Express Link, or CXL™, as the primary fabric for interconnecting a variety of processors, I/O resources, memory resources, and storage for data centers, hyperscalers, and similar computing clusters. Artificial intelligence is similarly soaring in popularity, and demanding massive data sets for the learning processes. It is likely to move into the next generation of automobiles as well as cars adopt data center technology. CXL allows mixing resources in a fashion that meets the needs of these systems, and flexibility for each end user to deploy a different mix.

CXL uses a light-weight protocol running on industry standard PCIe buses that adds moderate latency to the overall connection. This aspect of CXL is essential to consider memory expansion on CXL. However, there are some tradeoffs that must be considered, and CXL solutions proposed that use DDR DRAM for the memory media exhibit some worrisome characteristics that have system architects concerned in ways that may slow or limit adoption.

CXL Native Memory™ addresses many of these concerns with CXL memory expansion by correcting some of the legacy baggage of typical memory solutions.

CXL Addresses the Memory Wall

The "memory wall" is the gap between the demands for data storage and the ability of DRAM technology to meet that demand. The days of quadrupling per-DRAM capacity every 3 years has stretched out to 12 years or more due to the difficulties of making DRAM in ever finer processes.

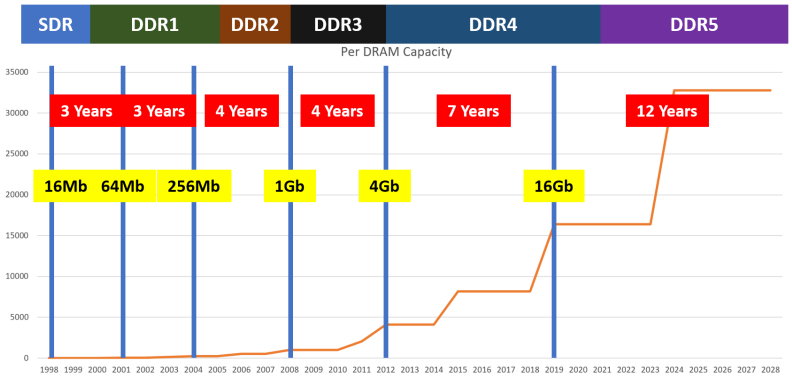

Figure 1: DRAM Per-Die Capacity Trends

The next big brick in the memory wall is the inability to connect DRAM modules (DIMMs) directly to the host CPU. DDR1 allowed up to four DIMMs on each memory channel up to the 400 Mbps data rate of the day. When DDR2 came along, the increase in data rate to 800 Mbps was welcomed, but the consequent reduction to three DIMMs per channel forced addition of a second memory channel in order to

increase memory capacity. Unfortunately, this trend hit the obvious wall when DDR5 went to one DIMM per channel. At roughly 300 pins per channel, it became difficult to add more channels on the CPU to increase capacity, and as a result, DDR5 offers no advantage over DDR4. Some systems are proposing to go to 12 or even 16 channels, however physical space limitations on motherboards creates other problems.

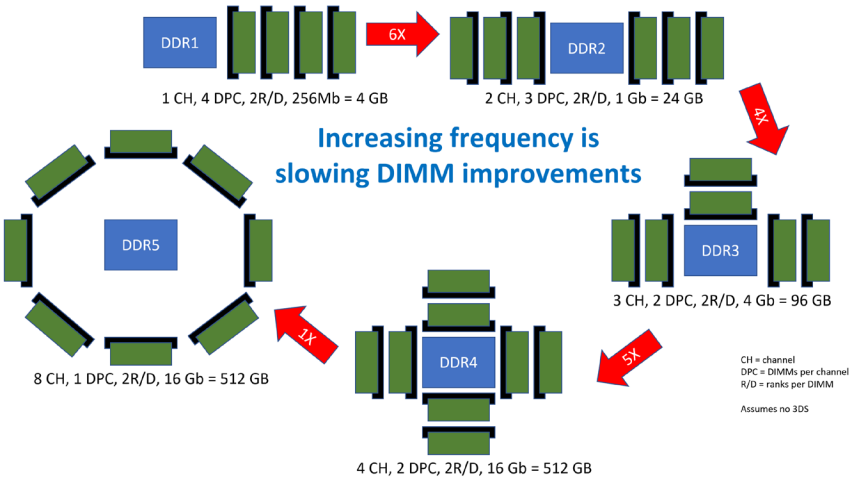

Figure 2: DIMMs-Per-Channel Trends

Compute Express Link, or CXL, features a low pin count method for expanding system resources. A DDR5 DIMM offers two 25.6 GB/s subchannels over those 300 pins, each of which is half duplex (reads and writes cannot occur simultaneously). A CXL memory module with a x8 PCIe configuration, however, operates over roughly 32 active pins and gives a performance of 32 GB/s, plus operates in full duplex mode (reads and writes simultaneously).

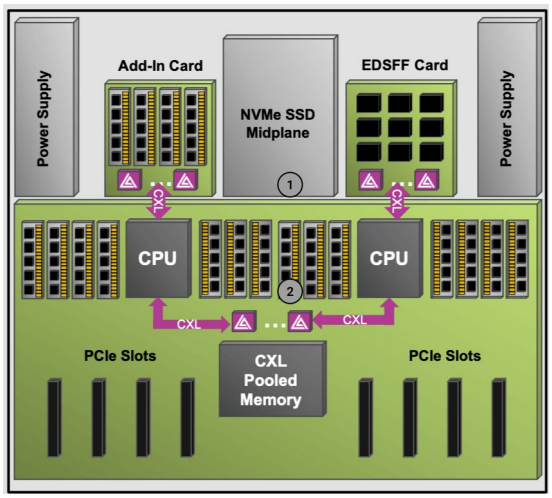

Figure 3: CXL-Based Server Architecture

DDR DIMMs and CXL memory modules are not mutually exclusive; emerging systems are likely to use both in order to achieve the highest system memory capacity. With only 32 active signals to route, combined with CXL's ability to add switches for

channel expansion, memory capacity thousands of times higher than DIMM-only solutions may be realized.

The CXL protocol is non-deterministic, which in this context means that the time from the memory controller issuing a read or a write command until the time the command is completed is variable. This contrasts with the DDR5 DRAM protocol where the timing of these commands and others are completely under the control of the CPU memory controller and fixed by DRAM fundamental timing. This distinction will be an essential aspect of calculating parameters such as latency. However, non-determinism also allows CXL-based memory solutions to add value to the system by offloading requirements like refresh, post package repair, data scrubbing, or other features typically executed by the CPU memory controller and firmware.

CXL Memory Challenges

If it sounds good so far, it should. CXL memory expansion is helping address the needs of artificial intelligence applications and other in-memory compute programs that demand ever-increasing amounts of high performance memory. However, with great performance comes great responsibility. The following concerns with CXL modules with DRAM for memory expansion impacts the adoption of this feature:

1. **DRAM Power**: DRAMs are power inefficient, and adding more DRAM means more power invested.
2. **Module Power**: CXL's serial interface adds additional power consumption.
3. **Security**: DRAM has inherent security risks that CXL may make more difficult to track.

Wolley CXL Native Memory provides a unique solution for CXL memory expansion that addresses each of these concerns.

Challenge: DRAM Power

One of the reasons that DDR DRAM consumes so much power is that the devices are designed to meet thousands of applications and therefore are not tuned for any of them. For example, a CXL Memory Module typically connects a rank of ten DRAMs each with a x4 I/O interface to form a 40-bit wide channel (32 data bits and 8 bits of ECC; metadata requirements must be merged with ECC).

As shown in Figure 4, each DRAM in a rank duplicates functions also in each of the other DRAMs. Refresh counters, for example, are designed into each DRAM even though they all operate at the same time when a refresh command is issued to a rank. Each DRAM implements a Per-Row Activation Counter (PRAC) to detect rowhammer attacks even though all ten DRAMs get exactly the same row activation.

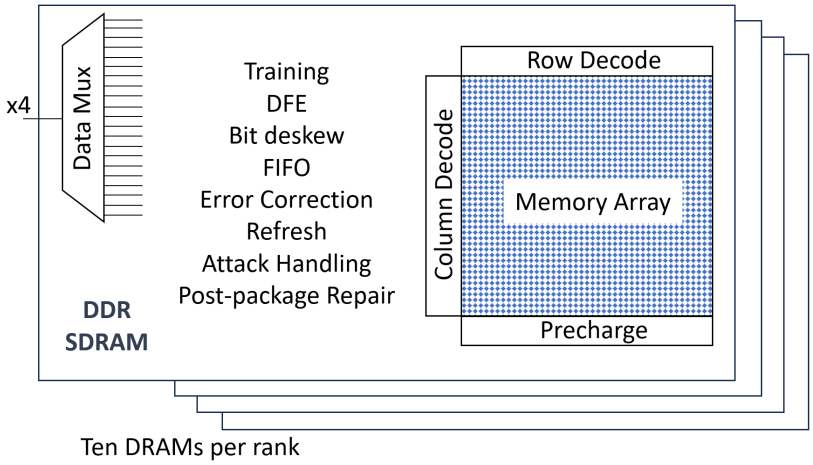

Figure 4: DDR DRAM Functions

CXL Native Memory as shown in Figure 5 is a flow control unit (FLIT) based solution. CXL FLITs are packets containing 64 data bytes (512 bits) that match the natural size of one processor cache line, so naturally, a DRAM with a 512-bit wide memory port allows transfer of one FLIT on each clock cycle. This slows the clock rate, lowering power and improving efficiency.

CXL Native Memory deploys a wide I/O memory device with centralized memory control logic, consolidating all the DRAM features scattered across many devices into one. The interface between the logic chip and memory chips is direct silicon-to-silicon connectivity, allowing very low power drivers. This consolidation reduces memory power significantly, at least 50%.

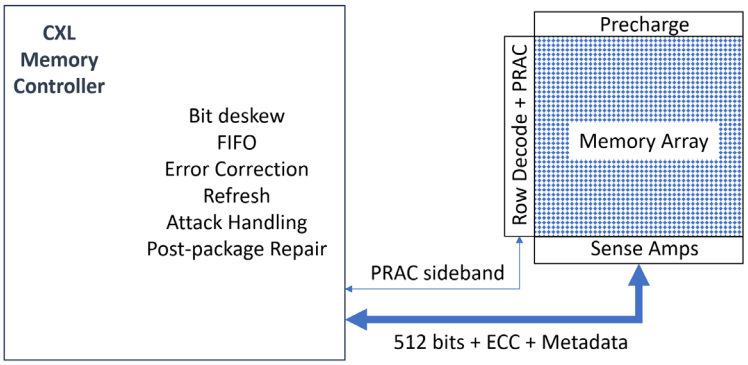

Figure 5: CXL Native Memory Centralized Control Functions

DDR has tried to offer a partial array self refresh (PASR) concept in the past where blocks of the DRAM not in use could have self refresh turned off. This is not included in current generation DDR5, however CXL Native Memory not only adds this feature, but the implementation is extended to runtime activity and merged with the CXL system memory allocation scheme used for memory pooling as reflected in Figure 6. Memory that has never been allocated, or has been freed from use, does not need to be refreshed. CXL Native Memory turns off refresh for unused memory regions automatically. This results not only in additional power savings, it also results in higher performance since memory access interruptions are reduced.

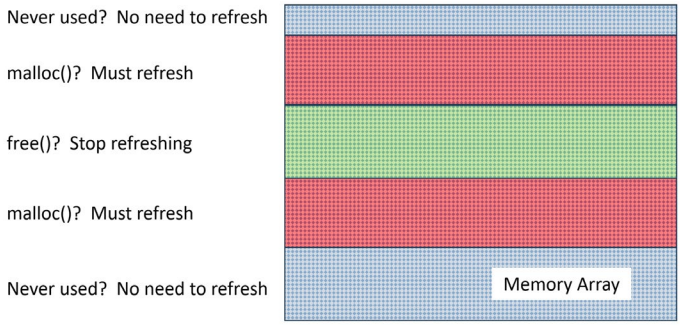

Figure 6: CXL Native Memory Refreshes Only Cells in Use

Metadata is a hot topic in computer memory design. Metadata allows the system to store "hidden" information about the memory regions, and the amount of metadata desired is increasing. Standalone DDR solutions imply that adding metadata would require more DRAMs, i.e., an 11th DRAM in each rank, which would clearly increase the cost of the memory solution and the pincount. CXL Native Memory allows addition of metadata as in Figure 7 which is covered by the ECC codes, and each bit of metadata added only increases the die size by 0.2%, so customers can define the number of bits they need for their application. CXL FLITs include bits of metadata that can be included in these arrays.

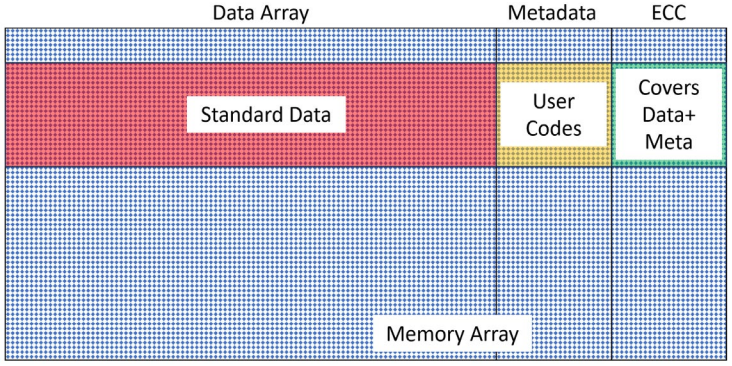

Figure 7: CXL Native Memory Array with ECC Coverage on Metadata

One can easily recognize that ECC coverage of one device with a 512-bit word is much more efficient than 10 DRAMs with ECC coverage on only 128-bit words. DDR solutions incur a 12.5% die penalty for ECC to get single-bit correction. CXL Native Memory can support a superior 3-bit error correction scheme with a 6.4% die adder, i.e., better protection at lower cost.

Challenge: Module Power

CXL Memory Modules are designed for plug-in cards using EDSFF outlines such as E3.S that support hot plug into large external chassis such as in Figure 7. As a result, the PCIe interfaces must support long cabling which requires very high power.

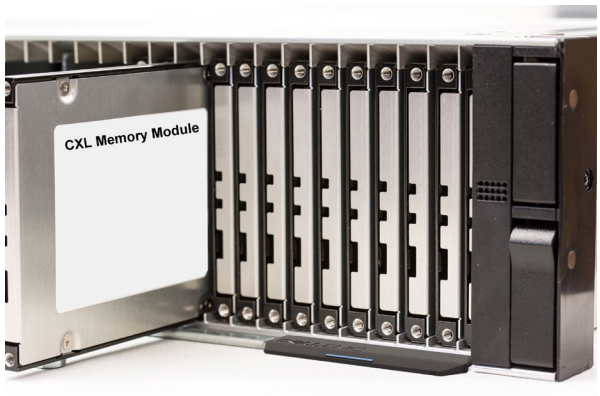

Figure 8: CXL Memory Expansion Chassis

CXL Native Memory reduces the PCIe interface power by assuming a short-throw point to point interface, either with solder-down BGA or with small socketed modules that reside on the main motherboard. This reduces the solution power even lower the popular LPDDR (low-power DDR) memories in use in notebooks today.

Feature	CXL Native DRAM 8-lane PCIe Gen5	LPDDR5 2x LPDDR-6400 x16
Bandwidth	32 GB/s each direction (Full Duplex)	25.6 GB/s one direction (Half duplex)
Power	2.0 pJ/bit	4.0 pJ/bit
Active Signal Count	32	136

CXL Memory Modules that use standard DDR memories have another power problem: most of the DDR overhead such as training and error correction are duplicated in the controller as well. Compare Figure 9 and also Figure 4. This not only costs in terms of power burned, this also costs in terms of silicon purchased.

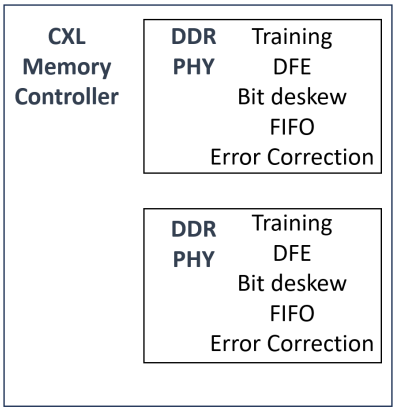

Figure 9: Standard CXL to DDR Memory Module Controller Logic

By combining the DRAM control logic in a single device, and eliminating the need for complex interface training, CXL Native Memory much more efficiently processes

DRAM control functions with lower power and higher performance. This optimization is shown in Figure 5.

Challenge: Security

Rowhammer security attacks have become a major challenge for DDR-based memory solutions where a virus program repetitively activates a row of memory with the intent to use DRAM crosstalk to inject virus code into adjacent memory locations. The standard solution, Per-Row Activation Counting (PRAC), uses counters to determine when the attack is occurring then invokes refresh operations on the victim rows of the DRAM core. This attack is visualized in Figure 9.

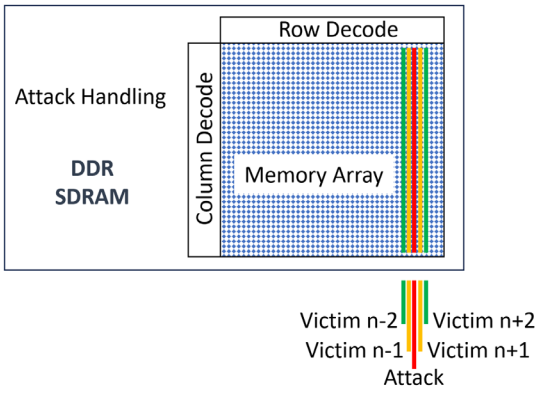

Figure 10: Rowhammer Security Attack

CXL Native Memory also supports PRAC, however the implementation is fully integrated into the partial array refresh capabilities. If the attack is performed on an unallocated memory region where the victim rows are not used, it can be ignored. If the victims are in a valid region, the CXL Native Memory Controller can suspend subsequent access to that region and perform error check scrubbing on the attacked rows. Error reporting to the host regarding the rowhammer attack is performed over the CXL.io interface and statistics are maintained in the CXL Native Memory device.

General Purpose Acceleration

CXL offers another very useful feature. Where the CXL.mem part of the protocol addresses direct memory addressing well, CXL also requires CXL.io, a protocol based on the PCIe interface. This back channel into the controller can be very useful in that commands not supported with DDR are now possible.

One example is the ability to perform a memory fill operation. Currently, with DDR, there is no mechanism to tell a DDR DRAM to fill address range X to Y with a pattern. This is because such a command would break the deterministic nature of the DDR protocol. However, CXL's non-deterministic protocol allows such commands, and the CXL.io interface is a great place to add them.

The utility of memory fill cannot be underestimated. When an operating system allocates a new block of memory, it clearly cannot let the requesting application see what data was stored there in case there were passwords, encryption keys, etc. A "fill block with 0" operation is a great acceleration over looping through that memory to write the new value. The CXL Native Memory implementation can also include the user's metadata and do the ECC calculations directly on the module without engaging the processor or the CXL bus.

Naturally, the CXL.io back channel allows for many new functions that can be added over time; compression, encryption, processing-in-memory, and so forth may be added to the roadmap or into customer-specific solutions.

CXL Performance Comparison

The PCIe interface adds latency to each access. Many have concerns that this latency will negatively impact system performance, and in fact for any given program thread this is partly true. However, this overlooks a number of factors. The first factor is that single-core processors are increasingly rare, and most systems run many cores simultaneously. This both increases the traffic and the randomness of the memory accesses. The second factor is the distinction between DDR half duplex and CXL full duplex operation, which in some cases can double total throughput. Figure 11 shows the results of a simulation run using common performance benchmarks. Under light load, the left part of the X-axis, the LPDDR shows a very small improvement over CXL. However, as the load increases to the right, the bubbles caused by the DDR half-duplex nature begin to dominate and while CXL performance stays nearly flat, DDR performance decreases significantly.

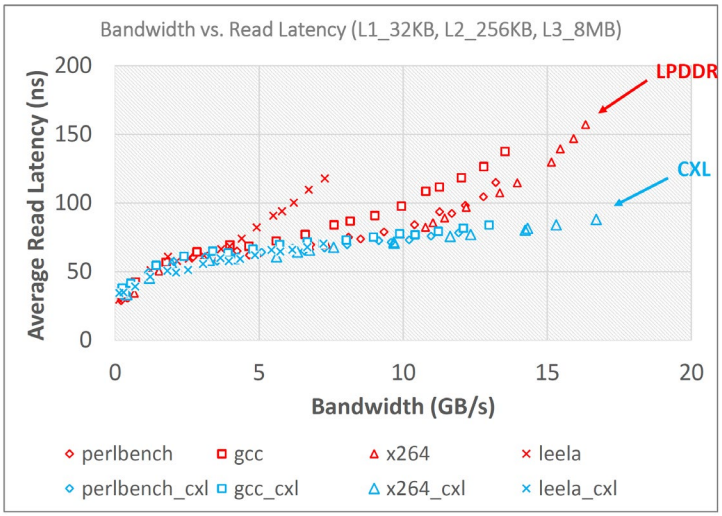

Figure 11: Performance of CXL Native Memory Versus LPDDR

CXL Native Memory for Artificial Intelligence (AI)

A variation of DDR called High Bandwidth Memory (HBM) has become the norm for artificial intelligence and machine learning applications. HBM provides good bandwidth, however due to the nearly 2000 pins for the interface, HBM requires a silicon interposer to connect HBM to the processor less than 2 mm away as in Figure 10. As a result, solutions are typically limited to no more than six HBM stacks. Even with expensive die stacking, this limits the memory to 96 GB of local memory.

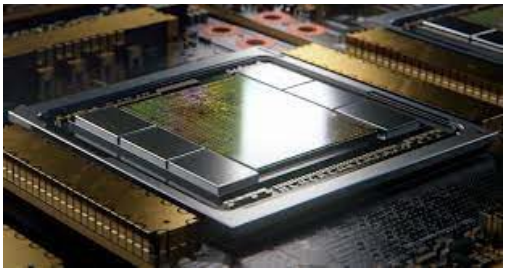

Figure 12: Typical AI Solution Using HBM

An increasing number of AI applications, such as Large Language Models (LLMs), require much more than 96 GB. A typical data set is 140 GB or more, meaning that a lot of paging must occur between the HBM and external memory. The roofline model shown in Figure 11 highlights how adding memory raises the performance headroom of the solution before changing from memory bound to compute bound.

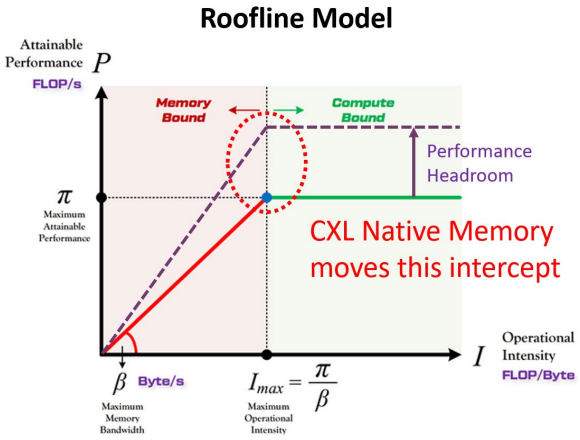

Figure 13: Roofline Model for LLM Performance

CXL Native Memory provides a simple expansion mechanism for AI solutions. With the low pin count, i.e., 32 active signals, and reasonably long throw allowing 100 mm between processor and CXL Native Memory module, incremental memory may be added to increase the total memory footprint. Each CXL Native Memory module may contain 8 GB of memory, so adding six modules in addition to the HBM would allow full sized LLMs to be stored locally with 384 GB/s aggregate throughput in full duplex operation. Since each CXL Native Memory module operates independently, any or all may be put into deep sleep or turned off to save power when not in use.

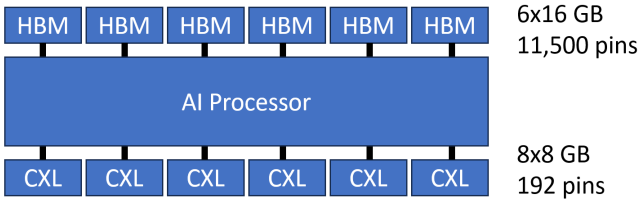

Figure 14: AI Processor with HBM and CXL Native Memory

CXL Native Memory for Automotive Applications

Automotive electronics have evolved into critical data center applications as cars become more networked, incorporate more sensors and controls, manage multiple high definition displays, and increase their artificial intelligence capabilities. Car electronics designs are already migrating to PCIe as the fabric connecting multiple high performance system-on-a-chip (SoC) processors. Each SoC typically has nine LPDDR DRAMs on a 144-bit data bus, and no shared memory. The redesign of the system replaces the nine LPDDR devices with three CXL Native Memories, and optionally adds more to the PCIe fabric for shared memory resources to simplify passing data between the processors. The new solution consumes far less power and takes up less space.

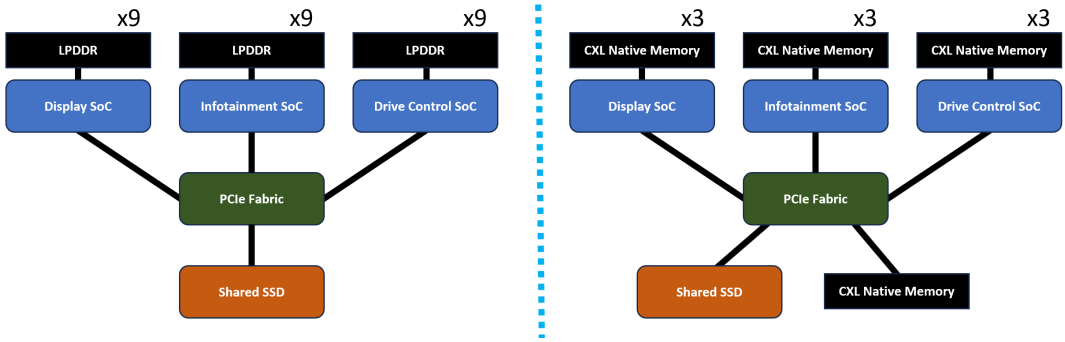

Figure 15: Automotive Solution with LPDDR Versus CXL Native Memory

Conclusion

CXL Native Memory rethinks the architecture of DRAM solutions by leveraging the increasingly popular CXL infrastructure. Replacing the DDR physical interface with a CXL interface allows memory transactions to match the 64-byte cache lines of processors via the 64-byte FLIT payload of CXL.

CXL Native Memory consolidates DRAM control logic currently duplicated in every DRAM with a single control logic chip. This allows better efficiency, and also enables functions not possible with discrete DRAMs. The result is a low power memory solution with a much simpler low pincount interface.

Obvious applications include notebook computers, AI accelerators, and industrial control systems, but increasingly, even server motherboard suppliers are wondering if this form of memory expansion helps them scale the memory wall.