Accelerating Innovation Using RISC-V and Gen-Z

Abner Chang, Mohan Parthasarathy, Michael Ruan
{abner.chang, mohan_parthasarathy, michael.ruan}@hpe.com

Gen-Z (https://genzconsortium.org/) is a scalable, universal system interconnect intended to simplify data access, simplify designs and solution architectures, unlock innovation and increase solution and industry agility, and reduce CAPEX / OPEX / NRE. Gen-Z uses a memory-semantic (read-write) protocol that enables multiple component types to efficiently communicate. Component types include processors, memory, storage, I/O, FPGA, GPU, GPGPU, DSP, etc. Universal communications simplifies component design and solution composition, making Gen-Z applicable to multiple solution types including clients, servers, storage, embedded, and message-based communications.

Figure 1: Example Conceptual Solution

RISC-V Processor with Integrated Gen-Z

RISC-V (https://riscv.org/) was created to be an open, free ISA (Instruction Set Architecture) intended to unlock processor innovation and increase processor solution and industry agility. Bringing these two innovative technologies together in integrated processor designs seems only logical. RISC-V’s large address space, secured privileged execution environment, and extensible ISA naturally aligns with Gen-Z’s scalability, built-in security and hardware-enforced isolation, and robust and easily extensible functionality. Further, there are multiple processor and accelerator innovation opportunities using Gen-Z, e.g., Gen-Z Memory Management Unit (ZMMU) to transparently support memory at any scale, PCIe® Enhanced Configuration Access Method (PECAM) and Logical PCIe Devices (LPD) to enable massive scale-out storage solutions without requiring yet another interconnect and gateway to be interposed between the processors / accelerators and their data, far Atomics to simplify multi-component coordination (e.g., data-centric and hybrid compute), buffer operations (large data movement with optional cryptographically-secured and encrypted communications), pattern / regular expression offload, DRAM emergency
back-up services, DMTF (http://www.dmtf.org) Redfish hardware and security management, Ethernet and HPC messaging over Gen-Z, etc. This paper will explore a subset of these to illustrate the potential for close Gen-Z and RISC-V integration.

![High Level Block Diagrams of RISC-V Processor-based Gen-Z integrated Solution](image)

**RISC-V Processor-Integrated Gen-Z**

As illustrated in **Figure 2**: High Level Block Diagrams of RISC-V Processor-based Gen-Z integrated Solution, Gen-Z bridge logic includes: Gen-Z Requester and Responder protocol engines, ZMMUs, PECAM, interrupt / event handling, data mover (messaging and large data transmission and receive off-load engines), and management blocks. For RISC-V integration, we propose leveraging the RISC-V Control and Status Register (CSR) mechanism using the RISC-V misa (machine ISA) CSR bit-25 (Z-bit, refer to **Figure 3**: RISC-V “Z” extension for Memory-centric Instruction) to indicate that the RISC-V core supports memory-centric extension ISA and Gen-Z, i.e., Gen-Z CSRs and operations are available only if the Z extension is set. Gen-Z RISC-V CSRs are designed to manage Gen-Z resources and logic. This approach simplifies hardware and enables software to perform all Gen-Z discovery and enumeration, configure the processor-integrated bridge and peer Gen-Z components (at any scale), support event handling and error recovery, etc. In small-scale solutions such as a single enclosure, system firmware can provide all management services using Gen-Z in-band management packets or through out-of-band interconnects. In larger-scale solutions, a Gen-Z fabric manager may work in conjunction with resource managers (e.g. composable memory, storage, or I/O manager) to provide all management services using in-band management.
RISC-V Processor Integrated Gen-Z Data Mover

A RISC-V processor can include Gen-Z data mover functionality to efficiently and reliably exchange data and messages using Gen-Z buffer operations and Write MSG operations, thereby eliminating the cost and complexity of discrete NICs. A data mover can support one-sided (direct data placement similar to RDMA but with fewer constraints and resources) and two-sided messaging (target anonymous destination buffer). Two-sided messages can support up to 96-bit Receive Tags to optimize MPI communications, as well as Embedded Read functionality to reduce receiver oversubscription and minimize data copies to optimize MPI and Ethernet over Gen-Z.

A RISC-V processor can contain multiple RISC-V cores (RISC-V Hardware Thread, HART) that support the RISC-V Interprocessor Interrupt (IPI). Each RISC-V core can be customized based design-specific needs, e.g., Figure 3: RISC-V “Z” extension for Memory-centric Instruction, width of integer, instruction cache size, data cache size, privilege mode, memory management and etc. A simple 64-bit RISC-V core with base integer (RV64I) and RISC-V Machine-mode only is suitable for near data processing and can be used to support Gen-Z Large Data Movement (LDM). To offload RISC-V application cores, one or more separate simplified RISC-V cores could act as Gen-Z data movers and provide additional value-add services typically found in discrete Smart NIC devices at a fraction of the cost, complexity, and power consumption.

Within a data center, RISC-V processor-integrated data movers can reliably exchange data without using a traditional network stack thus accelerating performance while reducing solution jitter and power consumption. To simplify data mover logic, Gen-Z supports two memory management units referred to as a Requestor ZMMU and a Responder ZMMU. ZMMU functionality can be integrated in the processor MMU or implemented within a separate IP block attached through the processor-internal coherency interconnect. RISC-V’s large physical address space (57+ bits) simplifies design and integration without sacrificing address reach or solution scalability. Additional details are found in the subsequent MMU section.

Gen-Z IO with PCIe Compatibility

I/O devices that support Gen-Z LPD (Logical PCIe devices) functionality can be discovered and configured using standard PCI system software using LPD (Logical PCIe devices) functionality. Each LPD appears as PCIe Root Complex Integrated Endpoints (RCIE). A PCIe RCIE is a simplified PCIe endpoint that requires very little software to manage.
(much of an operating system’s PCIe software is never invoked) and is not constrained by PCIe fabric rules. As a result, each processor or accelerator can support up to 8192 LPDs per supported PCI segment. If one LPD is provisioned per I/O device, then up to 8192 I/O devices can be supported which far exceeds the far exceeding the theoretical 256 device maximum permitted if using native PCIe (the actual number of PCIe devices is less as native PCIe solutions consume multiple bus numbers for PCIe switches and to accommodate hot-plug). This enables massive I/O scale-out solutions such as NVMe over Gen-Z storage without incurring the cost and complexity from deploying a NVMe over Fabrics gateway and another separate scale-out fabric. Further, all I/O components can be simultaneously shared enabling multiple processors / accelerators to access any NVMe storage device at any time at any scale with little to no coordination (e.g., single-writer, multi-reader paradigms). RISC-V can support LPDs on Gen-Z by implementing PECAM support in the Requestor ZMMU which translates PCIe configuration access to a Gen-Z component address. This enables an unmodified OS to transparently support Gen-Z LPD components and fully exploit Gen-Z’s numerous architectural benefits.

**RISC-V Processor-Integrated MCTP**

MCTP (Management Control Transport Protocol) is an open industry standard developed by the DMTF to exchange management data objects. MCTP operates over PCIe, I2C / I3C, Gen-Z, etc. making it the natural management technology to integrate into any component. Though MCTP has been broadly implemented in I/O devices, BMC (Base Management Controller), etc. an industry standard MCTP processor HCI (Host Controller Interface) is missing. This is an area where the RISC-V community can provide leadership in guiding the development of a MCTP HCI Figure 2: High Level Block Diagrams of RISC-V Processor-based Gen-Z integrated Solution and driving industry-wide adoption. MCTP processor integration will enable optimized Redfish-based management infrastructures, ease hardware development and deployment, unlock innovation and increase design agility, and deliver a consistent customer experience across all components irrespective of how they are interconnected. Further, data objects abstract the underlying hardware enabling new design innovation without fear of breaking software backward compatibility or gating solution time to market. For example, instead of developing interconnect-specific security mechanisms to authenticate components or to establish security sessions to support cryptographically-secured packet authentication and encryption, the DMTF is developing security data objects that can operate over any interconnect that supports MCTP. Developing and integrating a MCTP HCI into RISC-V processors will accelerate design development and broaden adoption.

**RISC-V Standard Extensions**

**RISC-V Standard Extensions for Gen-Z Security**

Integrated and flexible security is a key customer requirement for any new technology (See Gen-Z Component Authentication—Foundation for a Secured Infrastructure to understand some of the growing customer threats and motivation for strong infrastructure security). From the start, Gen-Z was designed with built-in trust and security features. Gen-Z supports strong packet authentication using HMAC (Hashed message authentication codes) to prevent packet tampering. HMAC integrity was combined with anti-replay tags to prevent bump-in-the-wire attacks. Gen-Z also supports data encryption services to ensure application privacy. Processor-integrated Gen-Z security features can complement RISC-V Cryptographic Extensions to provide an optimal secured infrastructure. At the time this paper was written, RISC-V Cryptographic Extension is proposed to be based on Vector Extensions for the standardization and secure execution of cryptography algorithms. This means RISC-V processor-integrated Gen-Z can utilize native RISC-V instructions to accelerate cryptographic algorithms (Figure 4 Utilize RISC-V Cryptographic for Gen-Z Security Features). Gen-Z packet HMAC generator could be integrated into the same RISC-V core which can remove software-generated HMAC overhead in a RISC-V Processor Integrated Gen-Z Data Mover. Dedicated
instructions provided by RISC-V Cryptographic Extension such as AES cipher algorithm and SHA-2 message digest enables fast and secured hardware HMAC code.

Gen-Z component authentication can be used to establish component trust and to mitigate In Situ malicious component insertion. DMTF-based component authentication mechanisms using MCTP over Gen-Z use strong cryptographic principles to detect counterfeit or compromised components and to verify firmware and component configuration. Each Gen-Z component is provisioned with a unique identity in the form of a cryptographic key pair and corresponding certificate hierarchy, rooted in a trusted Certification Authority (CA). The certificate is issued by Gen-Z component manufacture. Component authentication can be done by trusted firmware or software when a Gen-Z component is inserted using RISC-V Cryptographic Extension (Figure 4 Utilize RISC-V Cryptographic for Gen-Z Security Features). In addition to component authentication, strong packet integrity, and encryption, Gen-Z supports multiple hardware-enforced isolation techniques including destination component filtering, Access Keys to enforce component isolation and R-Keys to enforce page isolation to enable multi-tenant and virtualization solutions.

![Figure 4 Utilize RISC-V Cryptographic for Gen-Z Security Features](image)

**Gen-Z specific ZMMU operation**

Gen-Z supports two types of memory management unit (MMU) technologies—a traditional Page Table approach which can be integrated into a processor’s MMU for optimal performance (leverage the inherent caching logic and optimize latency) and a Page Grid which is discrete logic that can be accessed through the processor’s on-package coherency interconnect. A Page Table-based ZMMU can leverage RISC-V’s standard extension for supporting TLB-like PTE caching (similar to MIPS instructions such as tlbp, tlbr, tlbwi, tlbwr to operate its TLB). A Page Grid-based ZMMU is ideal for solutions that require maximum flexibility and scalability but do not want to modify an existing page table implementation. RISC-V Machine mode ZMMU instructions can be utilized by SFW to manipulate Z-structure.

**Gen-Z Atomics integration with RISC-V**

The Gen-Z specification enables atomic operations to be executed on any component. Processor atomic operations can be transparently transported across a Gen-Z fabric to another component for execution. This enables software to seamlessly operate across Gen-Z. Gen-Z Atomics provide a complete set of operations capable of supporting atomics used by multiple processor ISAs including RISC-V. There are 2 types of atomic operations—Near and Far. Near Atomics are those atomic operations which are executed within a processor. The target memory can be any directly-attached memory including DDR, HBM, or Gen-Z memory. Far Atomics are operations executed outside of the processor. Software maps addressable resources in one or more Responders through the ZMMU. Once mapped,
a processor can execute an Atomic operation which is transparently transported across the Gen-Z fabric by a Gen-Z protocol engine (PE) to be executed by the Responder component (this enables multiple Requesters or processor execution threads to communicate through a shared Responder with minimal complexity and performance impact). The Responder returns success or failure and any other applicable results. Any component type can initiate and / or execute Atomic operations in a Gen-Z fabric. The RISC-V Gen-Z interface can map multiple processor atomic operations onto the rich set of atomics provided by the Gen-Z to enable seamless software compatibility and optimal performance.

Far atomic support can be added to RISC-V to improve processor and accelerator communication and resource sharing efficiency. To avoid processor core stalls, RISC-V processors can use A-Extension support for the synchronization when multiple RISC-V harts are running in the same memory space, and Gen-Z Atomics across Gen-Z-attached memory, I/O devices, and accelerators. The design of memory-centric RISC-V far atomic memory instructions can be aligned with Gen-Z Atomic 1 OpClass, which enables Gen-Z atomic operations without processor’s intervention as illustrated in High Level Block Diagrams of RISC-V Processor-based Gen-Z integrated Solution. The mapping table of RISC-V A-extension (excluding LR/SC instructions) to the Gen-Z Atomic OpClass is delineated in RISC-V and Gen-Z Capability Mapping. Though currently RISC-V doesn’t support compare-and-swap AMO instruction (uses RISC-V “LD/SC” instructions), adding this atomic is under consideration by the RISC-V Foundation.

Instruction extensions can help to reduce the latency of load/store operations on the Gen-Z fabric. For general load/store semantics, CPU registers are the medium of memory movement for intrinsic data types like 32 ~ 64bits (un)signed integers. To accelerate the movement of non-intrinsic data structures like an array, instruction extensions are necessary to tell CPU when & how to collaborate with Gen-Z Data Mover logic to support optimized data transfers among Gen-Z components.
RISC-V Gen-Z MMU

Any component type can support a Gen-Z Requester ZMMU (used to translate a request such as a processor load or store instruction into a Gen-Z read or write request targeting a specific Responder such as a memory component) and / or a Responder ZMMU (used to validate and translate a Gen-Z read or write request to locate the corresponding Responder-specific resource, e.g., DRAM or persistent memory, which is then executed by the Responder). A ZMMU contains a set of Page Table Entries (PTEs) which describe the corresponding page. Gen-Z supports page sizes from $2^{12}$ to $2^{48}$ bytes enabling a wide range of design options. PTEs are programmed by trusted software such as system firmware or a trust OS sub-system.

A Page Table-based ZMMU-structure is located in local memory / cache, configured by system firmware at the initial POST stage, and protected by RISC-V Physical Memory Protection (PMP). This ensures that a ZMMU can be modified only in the RISC-V privileged execution environment, and any accesses in RISC-V lower privilege execution environments will fail.

A Page Grid-based ZMMU is a highly efficient alternative that requires fewer resources than a Page Table-based ZMMU as it does not require a hardware page table walker or tight processor MMU integration. A design may support multiple Page Grids with each Page Grid entry configured to support a single page size. This means the number of Page Grid-based entries in Page Grid table depends on the page sizes supported by a Gen-Z component. RISC-V Page-based 39 bit virtual memory is sufficient to support ZMMU page sizes which are smaller than 512GiB, while RISC-V Page-based 48 bit, 57-bit and up virtual memory are perfect for supporting large page size mapping.

Beyond 64-bit Addressing

In the future, will processors need to access more than $2^{64}$ bytes? The following sentence taken from RISC-V specification states, “It is not clear when a flat address space larger than 64 bits will be required. At the time of writing, the fastest supercomputer in the world as measured by the Top500 benchmark had over 1 PB of DRAM, and would require over 50 bits of address space if all the DRAM resided in a single address space” “Exascale systems research is targeting 100 PB memory systems, which occupy 57 bits of address space. At historic rates of growth, it is possible that greater than 64 bits of address space might be required before 2030.” RISC-V anticipates such a future and supports RV128 which provides a 128-bit flat address space. Gen-Z also anticipates such a future as a switch-based Gen-Z topology can support up to $2^{28}$ components and each component can support up to $2^{64}$ bytes of Data Space (application addressable resources) and up to $2^{52}$ bytes of Control Space (management, embedded executables and images, etc. resources). Perhaps such a future is closer than people realize.

RISC-V and Gen-Z Capability Mapping

Below table provides the additional information of RISC-V capability to Gen-Z features mapping.

<table>
<thead>
<tr>
<th>RISC-V Capability</th>
<th>Gen-Z Functionality</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open RISC-V Rocket Core</td>
<td>Gen-Z Functionality Blocks</td>
<td>The open source RISC-V core enables industrial community contributions to Gen-Z functionality blocks development on RISC-V processor integrated Gen-Z</td>
</tr>
<tr>
<td>RISC-V Open Instruction Architecture Set</td>
<td>Gen-Z Memory-semantic Operations</td>
<td>Open instruction architecture set is flexible for developing Gen-Z memory-centric and Gen-Z specific instruction such as ZMMU instructions</td>
</tr>
<tr>
<td>Variants of RISC-V Core (HART) in Processor</td>
<td>Gen-Z Data Mover Gen-Z Security Features</td>
<td>Standalone RISC-V core dedicate in Gen-Z Data Mover and Gen-Z security features remove the overhead from RISC-V application cores and software</td>
</tr>
</tbody>
</table>
### Conclusion

RISC-V processors and accelerators that integrate Gen-Z will:

- Be able to tackle new customer problems at any scale from the edge to the data center
- Be able to attack new data-intensive workloads with minimal software and hardware overheads
- Be used to construct fully composable, fully secured infrastructures
- Be used to deliver flexible and resilient data access
- Unlock innovation and increase solution agility
- Enable customers and the industry to optimize CAPEX, OPEX, and NRE