SNIA COMPUTE + MEMORY + STORAGE SUMMIT

Architectures, Solutions, and Community
VIRTUAL EVENT, APRIL 11-12, 2023

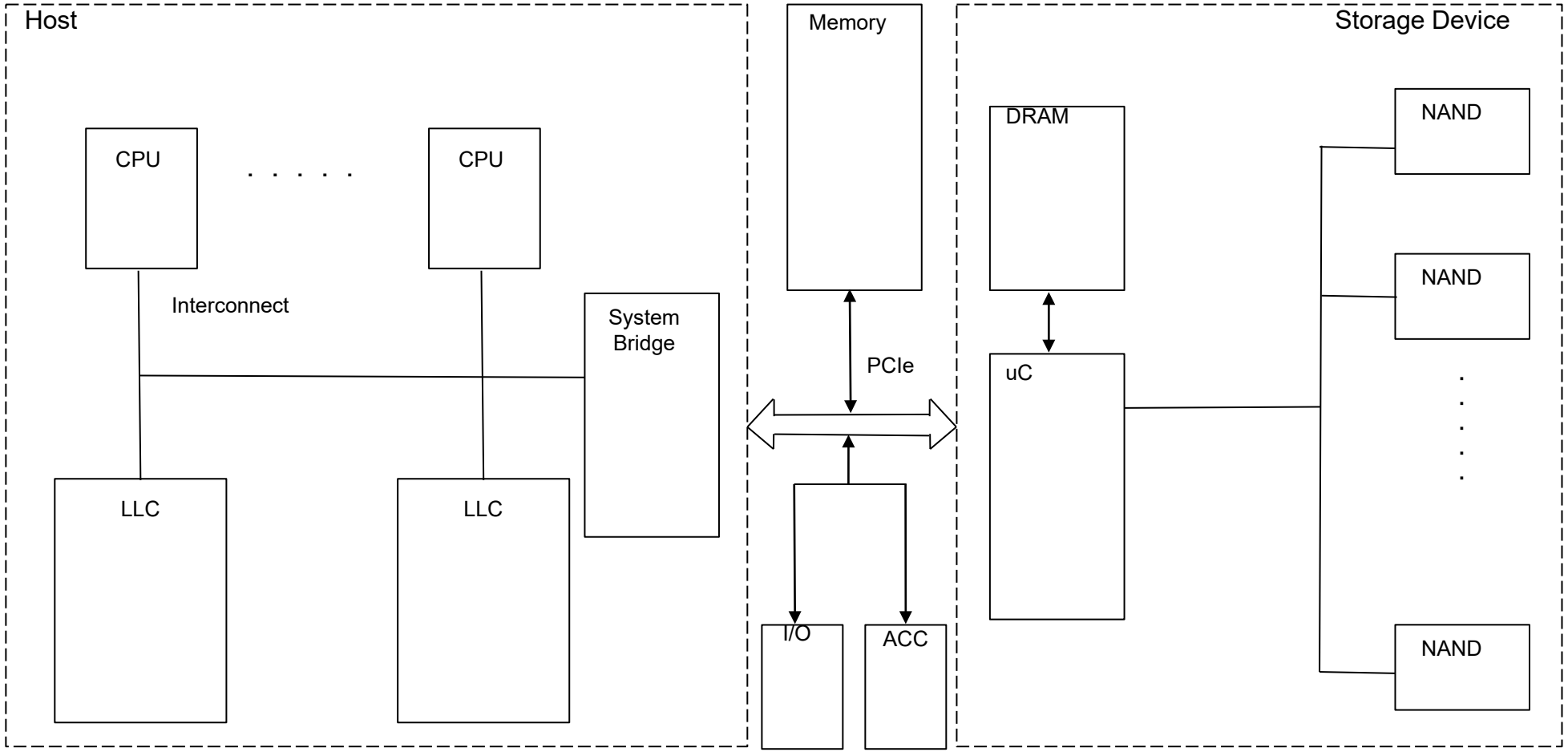
Accelerating Encryption for Solid State Storage Devices

Presented by Kelvin Goveas
Architect @ Azimuth Technology

Agenda

- Benefits of hardware acceleration for solid state storage devices
- RISC-V based hardware acceleration
- Hardware accelerator attachment options
- Hardware accelerator microarchitecture
- Status & future work

Block Diagram of Host with Storage Device



Key:

LLC - last level cache
ACC - crypto hardware accelerator
uC - microcontroller

Host Based Encryption

- Encryption can be performed by the host either by the
  - User through software libraries such as OpenSSL or
  - Kernel through APIs such as /dev/crypto on Linux
- Encryption throughput on the host can be improved by
  - Cryptographic instruction set extensions (ISEs)
    - x86 and ARM ISAs added crypto ISEs many years ago
    - RISC-V Zk* scalar 32/64-bit ISE ratified in Oct 2021 but not yet implemented in newer RISC-V cores
    - RISC-V vector crypto instruction spec Zvk* in public review but not yet ratified
  - Traditional crypto hardware accelerator (ACC) is attached to peripheral bus with DMA access to main memory

Storage Based Encryption

- Encryption can also be performed by the storage device
- Provides additional level of security for data that is not encrypted by the host
- Improves data security as keys are stored in the storage device rather than in main memory used by the host

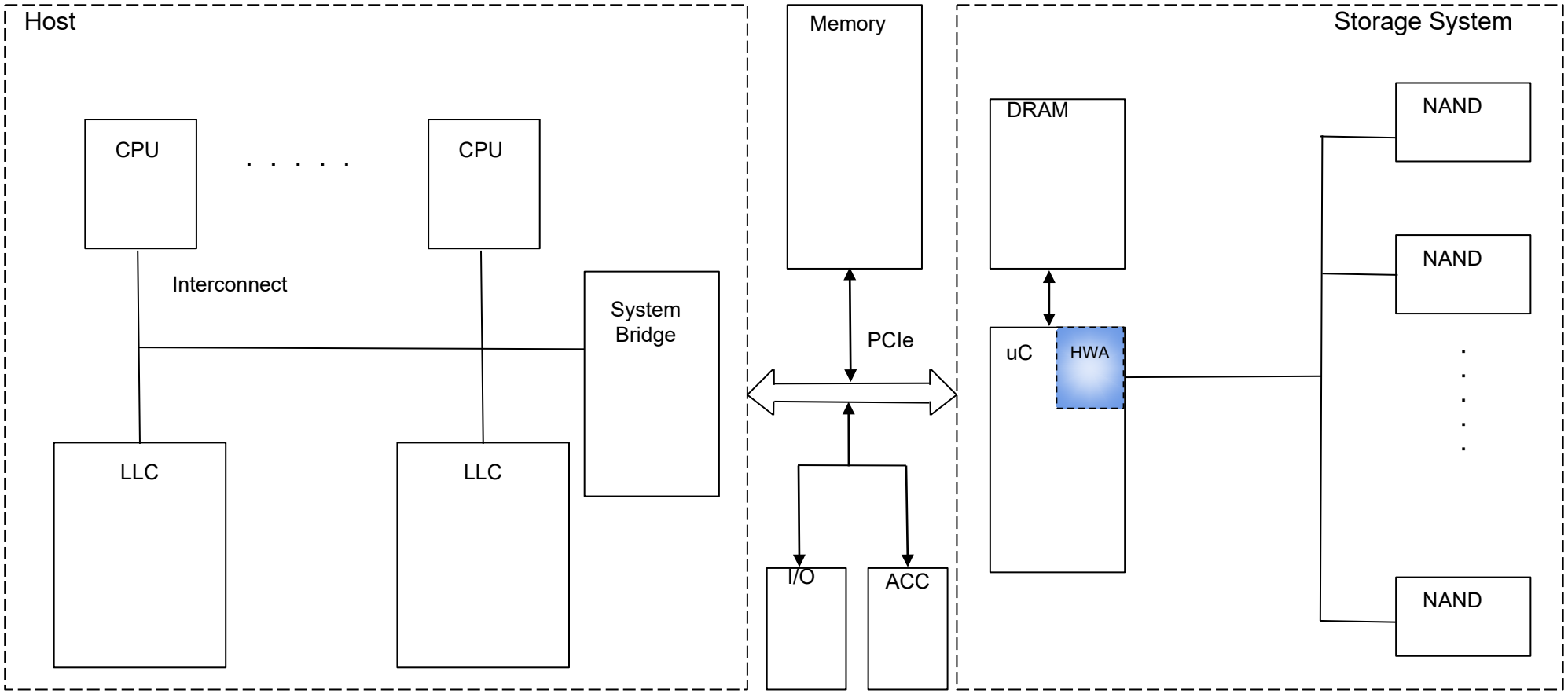
Hardware Based Storage Device Encryption

- Hardware solution can improve throughput and power efficiency of encryption/decryption of solid state (SSD) based storage devices
- Increases encryption read and write throughput relative to firmware library running on the microcontroller (uC)
- May also be faster than using crypto instruction extensions (ISEs) on the microcontroller
- Open source RISC-V ISA allows a hardware accelerator to be attached to a RISC-V based microcontroller (uC)

Hardware Accelerator for Encryption/Decryption

- Allows offload of encryption/decryption, data authentication and key generation from the storage device's microcontroller core(s)
  - Allows microcontroller to focus on flash translation layer (FTL) functionality such as logical to physical block address mapping, wear levelling and garbage collection
- Better power efficiency than general purpose microcontroller
  - Cuts down power dissipated due to fetch, rename, dispatch and scheduling (for out-of-order core) of crypto ops expended in general purpose core

Azimuth Technology Hardware Accelerator



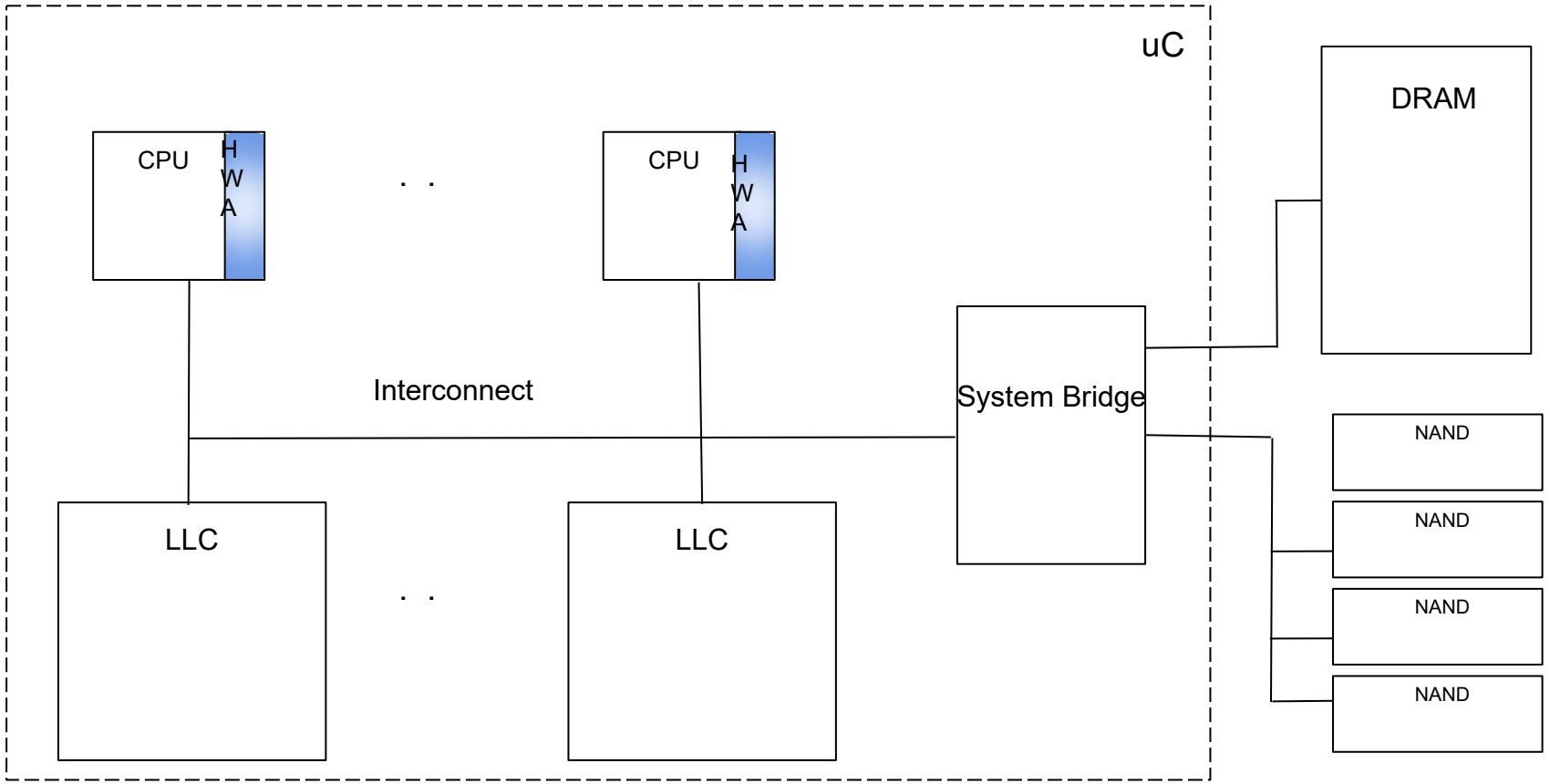
Azimuth Technology Hardware Accelerator

- Supports 2 different types of attachment
  - Tightly Attached
    - Can be attached inside the microcontroller CPU to the load/store unit or L2 cache
    - Only supported for RISC-V CPU with license which allows integration of an accelerator and/or addition of custom instructions by the customer
  - Closely Attached
    - Can be attached to the interconnect of the microcontroller SOC which interfaces to the last level cache (LLC)
    - Allows HWA to participate in a coherency protocol such as AXI or CHI as a non-caching node

Azimuth Technology Hardware Accelerator (cont'd)

- Either config allows firmware running on the microcontroller (uC) core to configure the accelerator (HWA) with a lower latency
  - Core initiates processing on the accelerator which then performs its own memory accesses and finishes computation
  - HWA shares the memory interface of the uC to perform
    - reads from solid state memory to decrypt data and write it to DRAM
    - reads from DRAM to encrypt data and write it to solid state memory

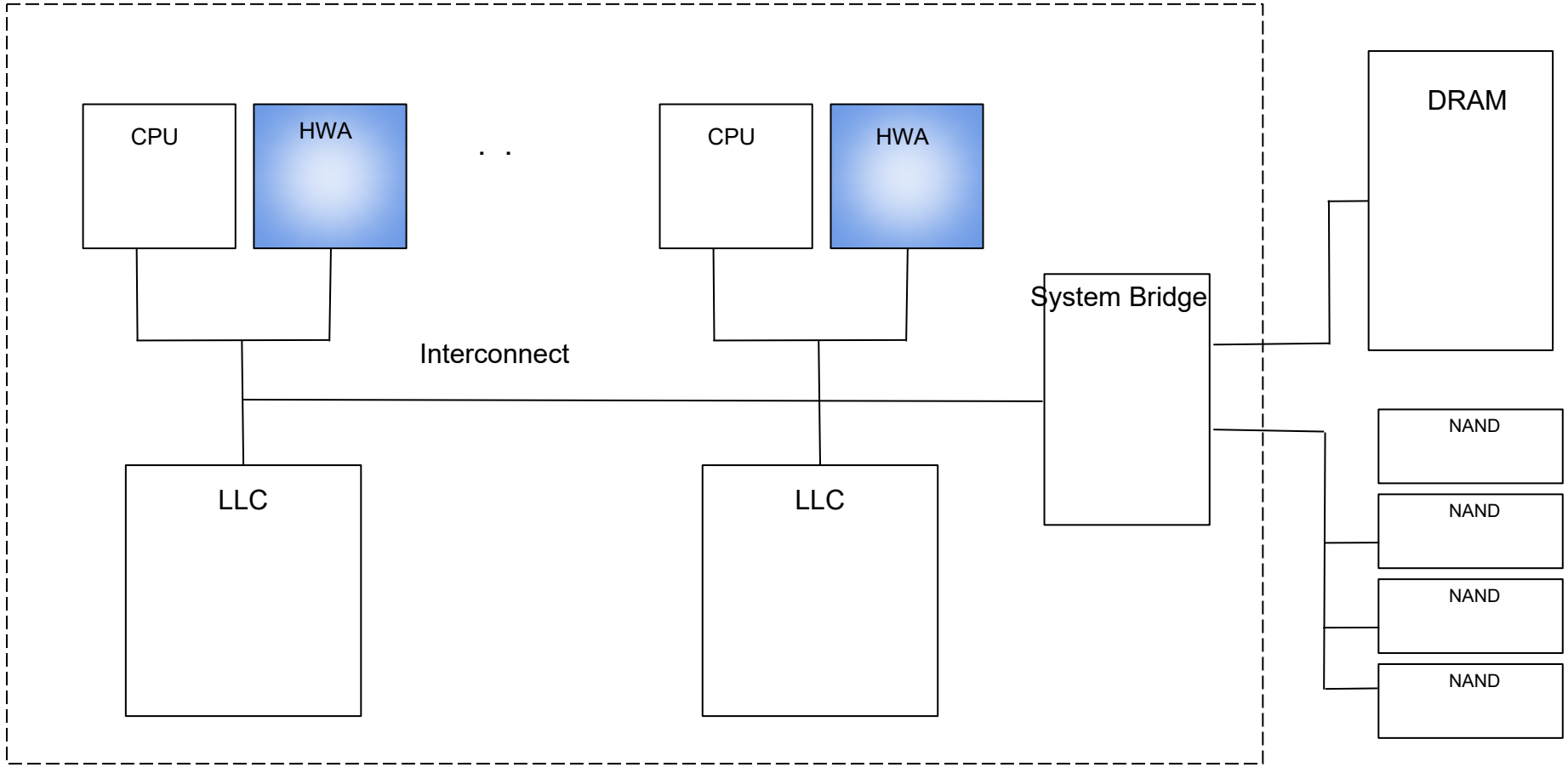
Tightly Attached Accelerator Config



Tightly Attached Accelerator Interfaces

- Management interface
  - firmware can use RISC-V special register (csr) read/writes to configure, start and stop HWA operation
- Memory interface
  - HWA can initiate reads or writes to the L1 data or L2 caches
- Completion interface
  - Synchronous - firmware polls for completion through management interface
  - Asynchronous - HWA can interrupt the core upon successful completion or exceptional result

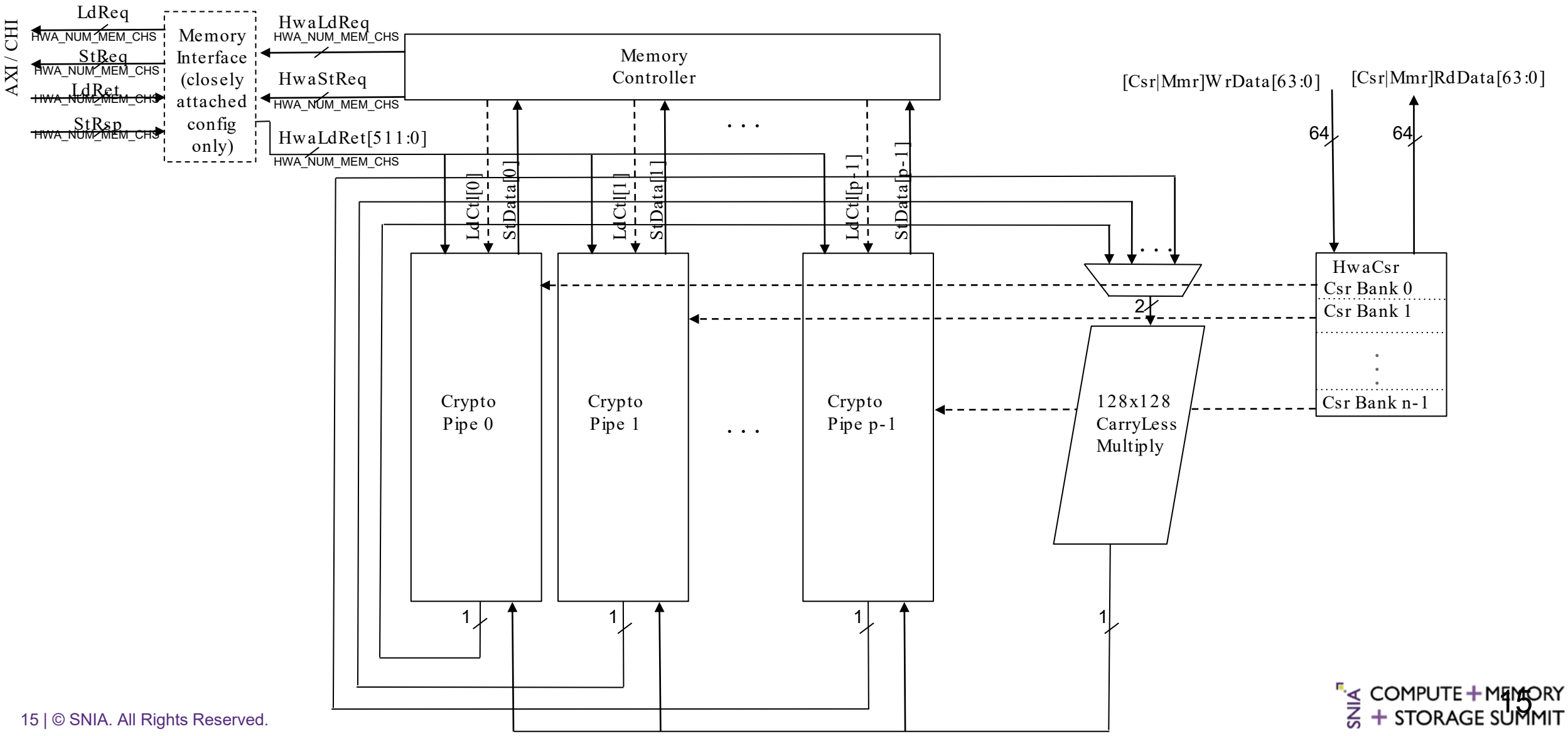
Closely Attached Accelerator Config



Closely Attached Accelerator Interfaces

- Management interface
  - similar to tightly attached config but uses memory mapped reads and writes instead of special register reads and writes
- Memory interface
  - HWA shares the memory interface of the core with the last level cache (LLC) which may use an industry standard AXI or CHI coherency protocol
  - HWA provides RN-I interface for CHI interconnect
  - Allows for encryption/decryption from memory directly into LLC which can be an L2, L3 or L4
- Completion interface
  - similar to tightly attached config where status may be polled or signalled through an interrupt

HWA Block Diagram



HWA Microarchitecture

- Support for processing separate threads by providing special register bank per thread
- HWA consists of ‘n’ pipes with shared memory interface and carryless multiplier where 1 <= n <= 8
- Banks can be mapped to pipes to allow software to configure processing throughput per thread
- Compute resources are decoupled from loads
  - only allocated on load return to improve compute efficiency

Encryption/Decryption Algorithm Support

- HW A optimized for authenticated encryption with associated data (AEAD), with confidentiality provided by using the Galois/Counter Mode (GCM)
  - AES used for the block cipher with authentication based on the GHASH
- The following encryption/decryption algorithms will be supported by the hardware accelerator
  - AES-128 (11 rounds), AES-256 (15 rounds)
  - SM4 (8 rounds)
  - SM3
  - SHA 256, SHA 512
  - AES128/256 GCM
  - SM4 GCM

Software Interface

- Hardware accelerator can be programmed through regular RISC-V instructions
  - only additional address space needs to be added for special register or memory mapped I/O accesses
- Memory region types can be configured using RISC-V physical memory attributes (PMA)

Configurability

- Configurability through parameters allows tuning of performance, power and area (PPA) to meet the constraints of a particular system
- Memory interface can be configured to be either AXI4 or CHI
- Number of crypto pipelines configurable to upto 8 pipes

Integration

- Offered as a soft IP which can be integrated into an SOC
  - RTL is coded using system verilog
  - Input parameters can be used to configure HWA
- Integration guide provided with each release

Status

- RTL, written in system verilog, currently in development using Amazon Web Services (AWS) EC2 platform
  - Starting point for datapath RTL was RISC-V crypto task group's open source RTL for 64-bit datapath which was expanded to 128-bits
- Unit level verification environment will use AWS EC2 F1 testbench but has not yet started
- Performance stats for AES-GCM will be obtained on AWS EC2 F1 FPGA platform

Future Improvements

- Update interrupt interface to support RISC-V AIA spec
- Improve hardware prefetcher to reduce memory latency
  - current prefetcher is configured by software and does not adapt to memory subsystem load
- Investigate susceptibility to differential power analysis (DPA) attacks and add support for any countermeasures
- Add performance counters to determine pipe utilization and memory throughput

About the Speaker

- Architect at Azimuth Technology, which was founded in August 2020
- 25+ year industry experience in design and development of x86, ARM and RISC-V CPUs
- Based in Austin, Texas, USA

Contact Information

Thank you for your time! Please contact us if you have any questions, comments or suggestions

- Email: azimuthtechusa@gmail.com
- Web site: www.azimuthtech.org
- LinkedIn Company Profile: Azimuth Technology
- LinkedIn Personal Profile: Kelvin Goveas

SNIA COMPUTE + MEMORY + STORAGE SUMMIT

Architectures, Solutions, and Community
VIRTUAL EVENT, APRIL 11-12, 2023

Please take a moment to rate this session.

Your feedback is important to us.

