

Summary [Up to 3 sentences]

Implement a KVS in the programmable switch, use it as a in network load-balancing write-through cache.

Exploit a theoretical result that caching only $O(N \cdot \log N)$ is sufficient to balance the load for a hash-partitioned KVS cluster with N servers.

Identify hot items by maintaining counters for each cached key, and a heavy-hitter detector (consists of a Count-Min sketch) for uncached key.

Challenge 1: Getting the hardware

- Often developed by *aaS companies
 - Little value in enlarging the ecosystem
 - Perceived value in keeping hardware proprietary
- Fear of support costs
 - Easy to support ASICs within a company
 - Much more effort for (even non-paying) customers
 - Liability, packaging, shipping, etc.



Challenge 2: Getting the documentation

- Some documentation is non-existent
 - Or merely embarrassing
- Lending hardware is legally easier than releasing documentation
 - There can be no standard NDA
- Documentation (or driver source) might reveal proprietary IP



Challenge 3: The big decisions have already been made



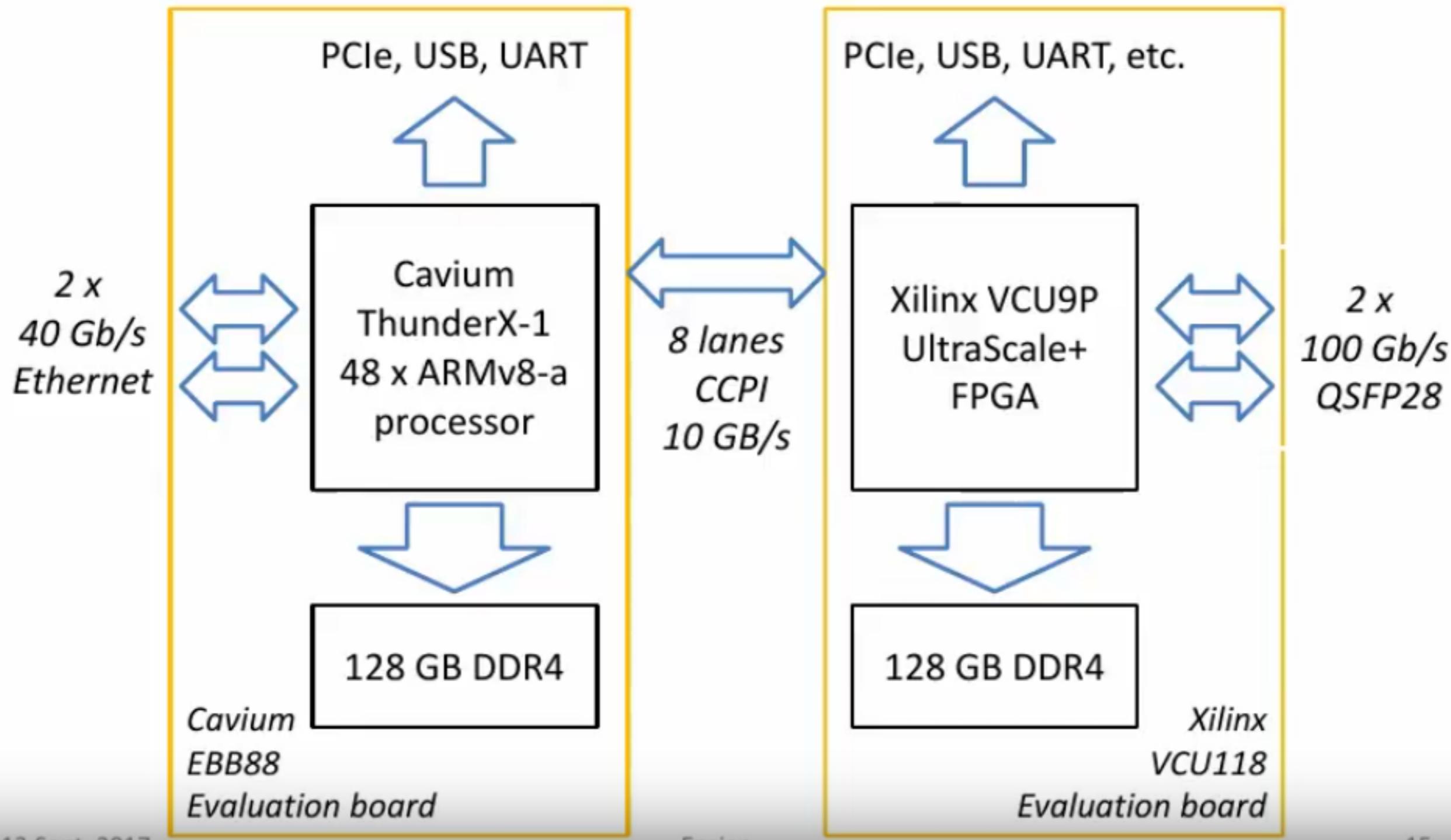
- Custom hardware is built with an application and mode of use already in mind
 - Cost-saving specializations
 - Particularly in drivers!
 - Creative abuse of such hardware is a limited
- Hardware probably never tested outside a narrow envelope
 - Researchers spend too much time finding bugs



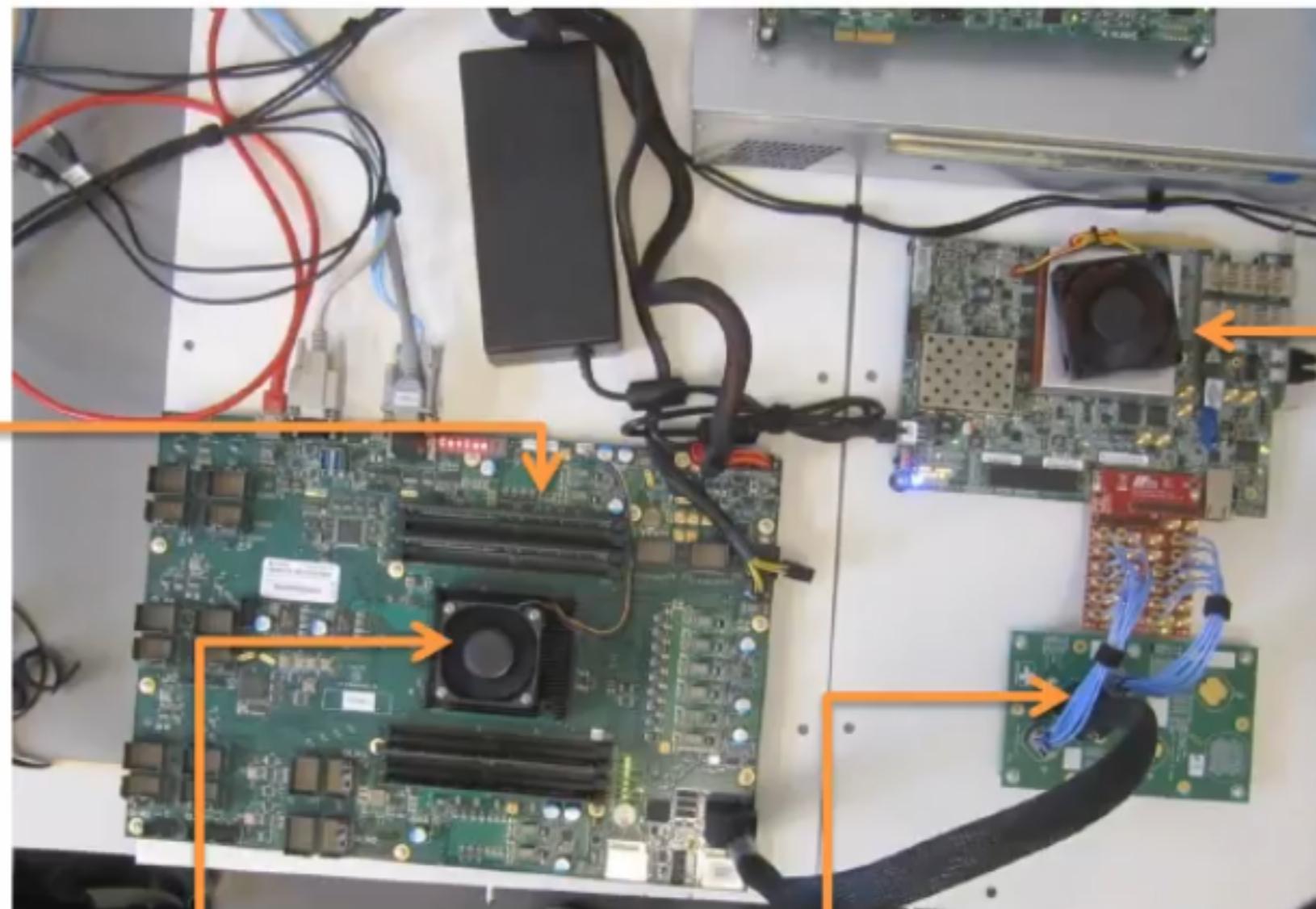
What if we had...

- A hardware *research* platform for system software
 - Massively *overengineered* wrt. products
 - Highly *configurable* building block for rackscale
- Perhaps we can *actually build it* at ETH...
 - Logical next platform for our research
 - Seed to other universities for impact

Enzian v.1



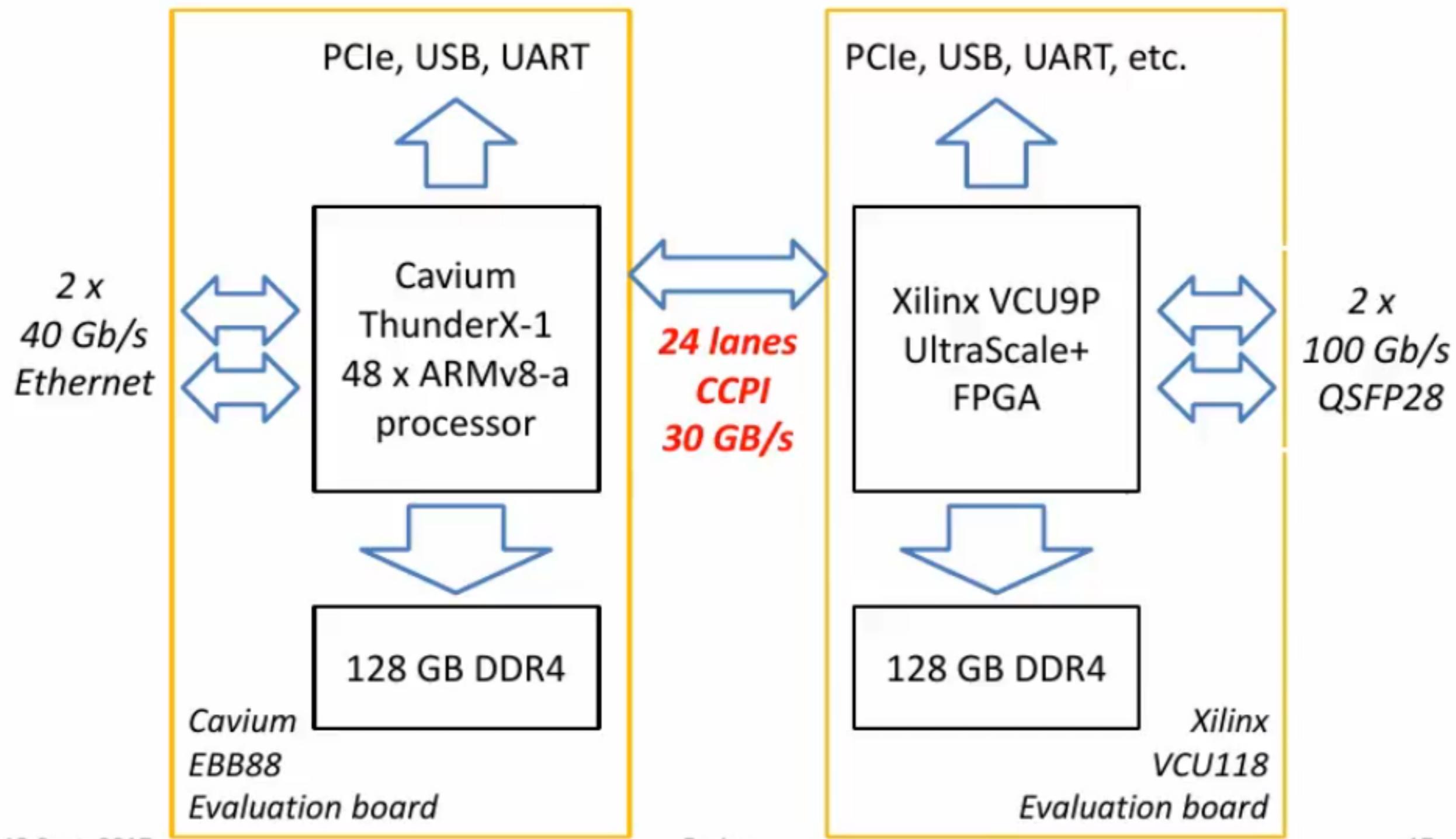
Enzian v.1



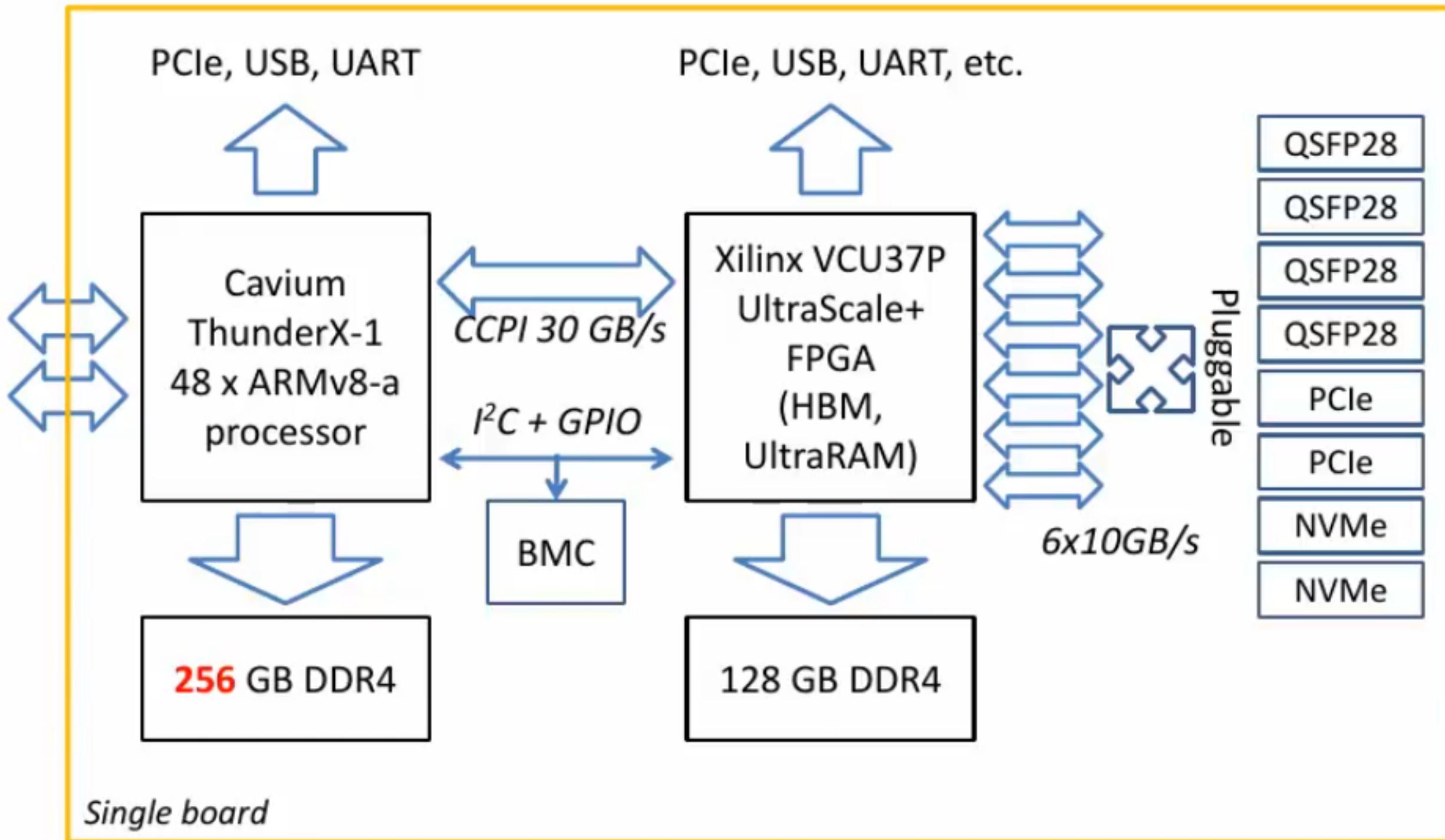
1 x Coherence link
(80Gb/s)



Enzian v.2 (October 2017)



Enzian v.3 (Q1 2018)





RESEARCH APPLICATIONS (THAT WE'VE THOUGHT OF)

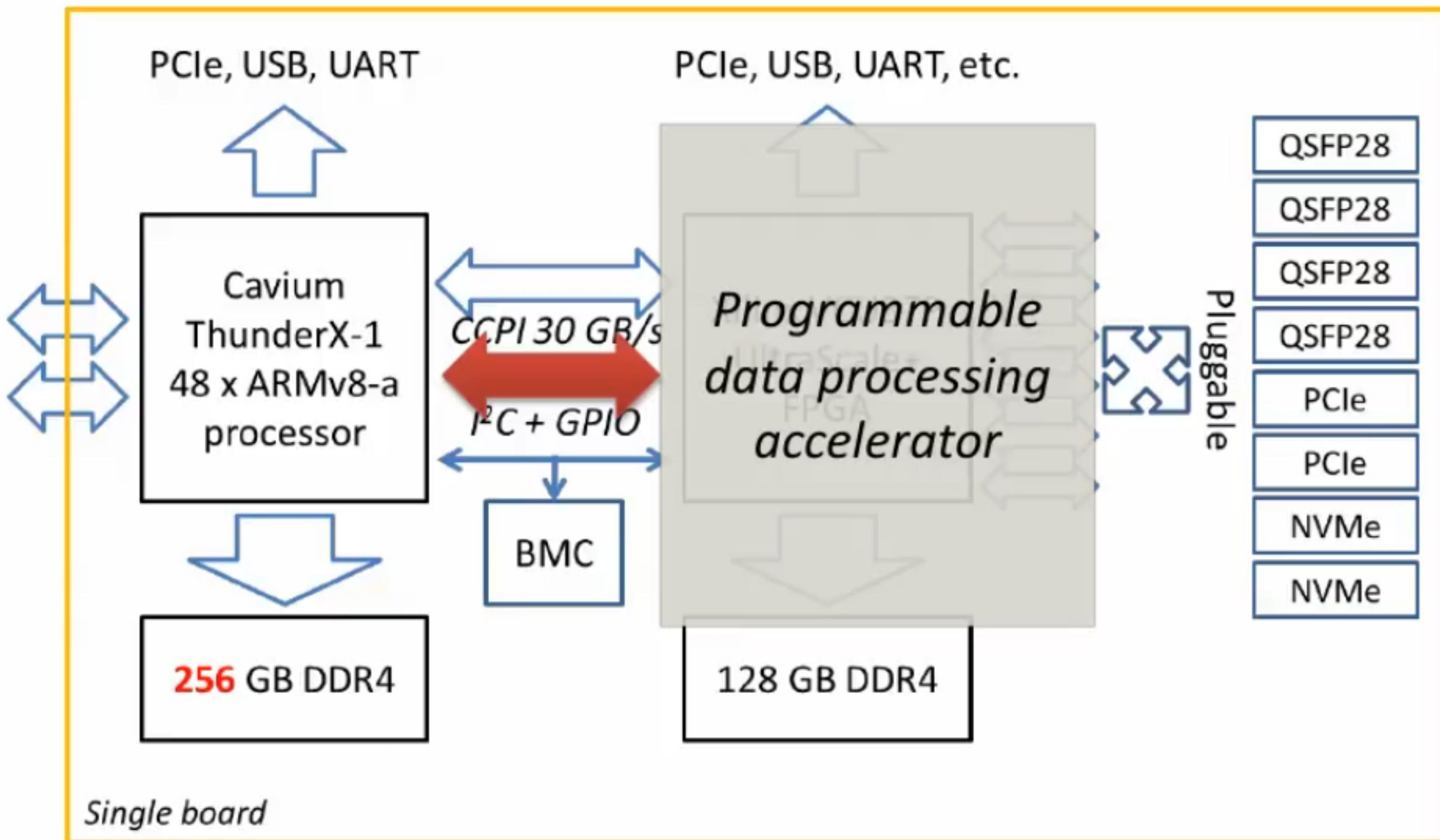


ENZIAN

Accelerator design



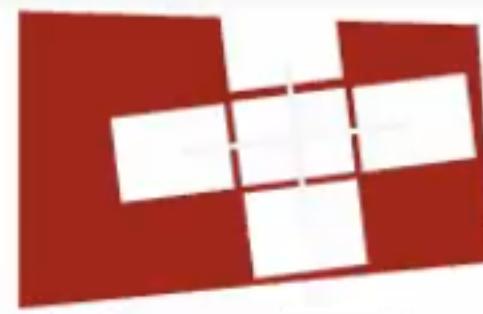
Systems@ETH Zürich



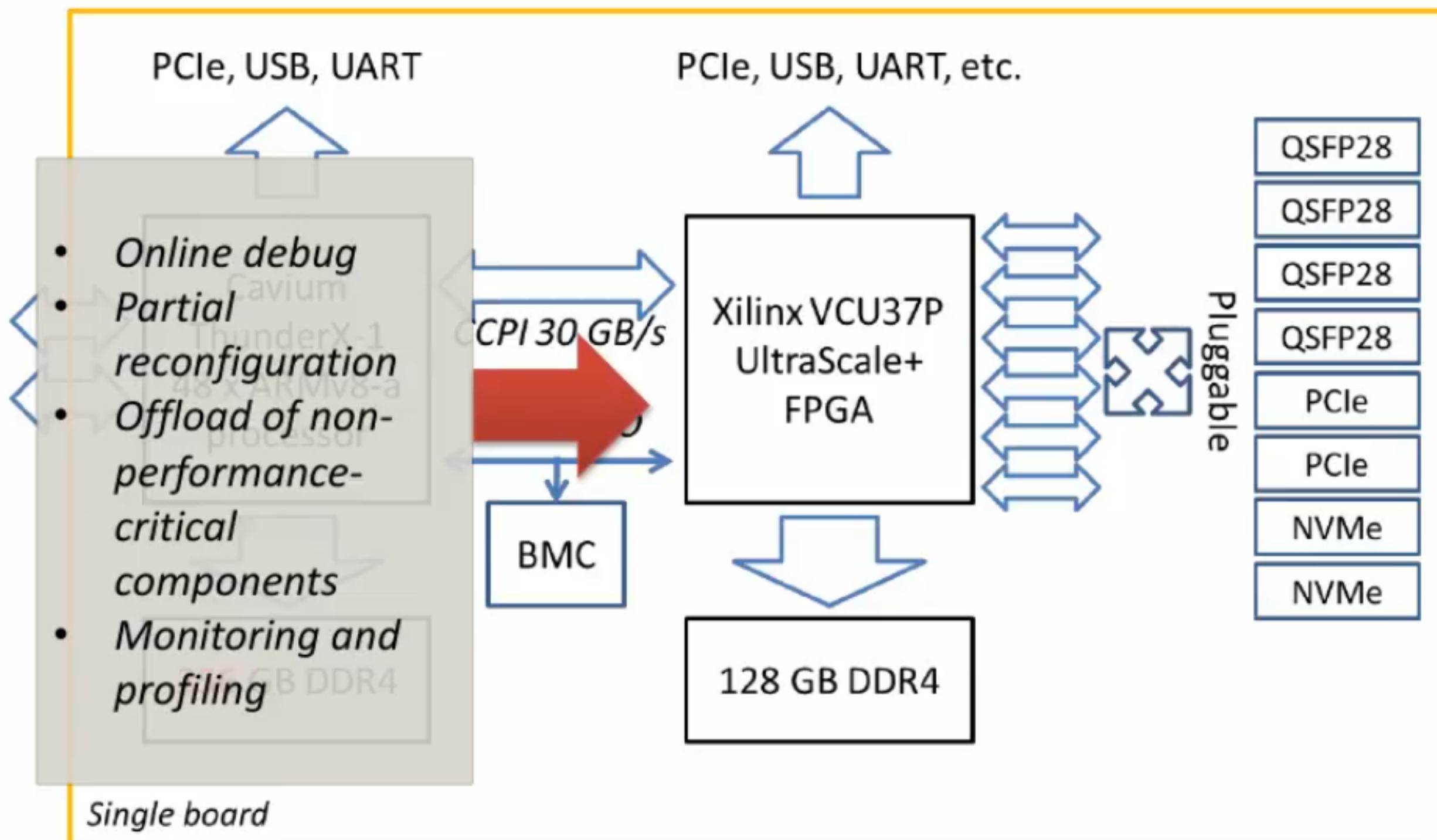


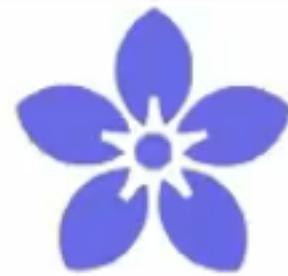
ENZIAN

FPGA board support



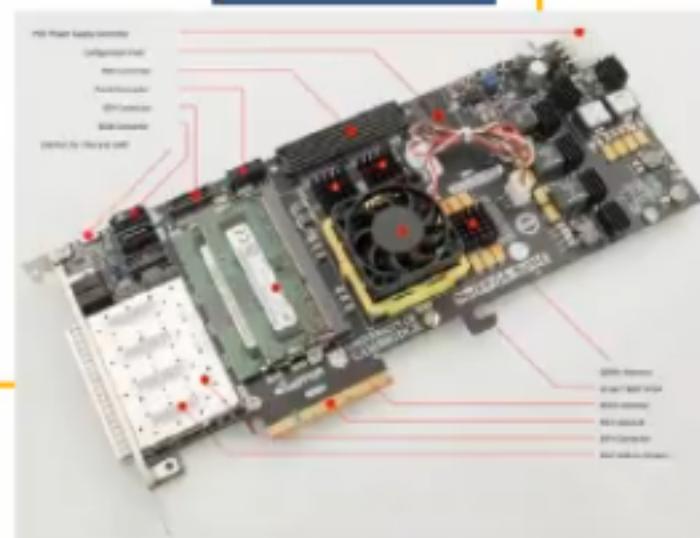
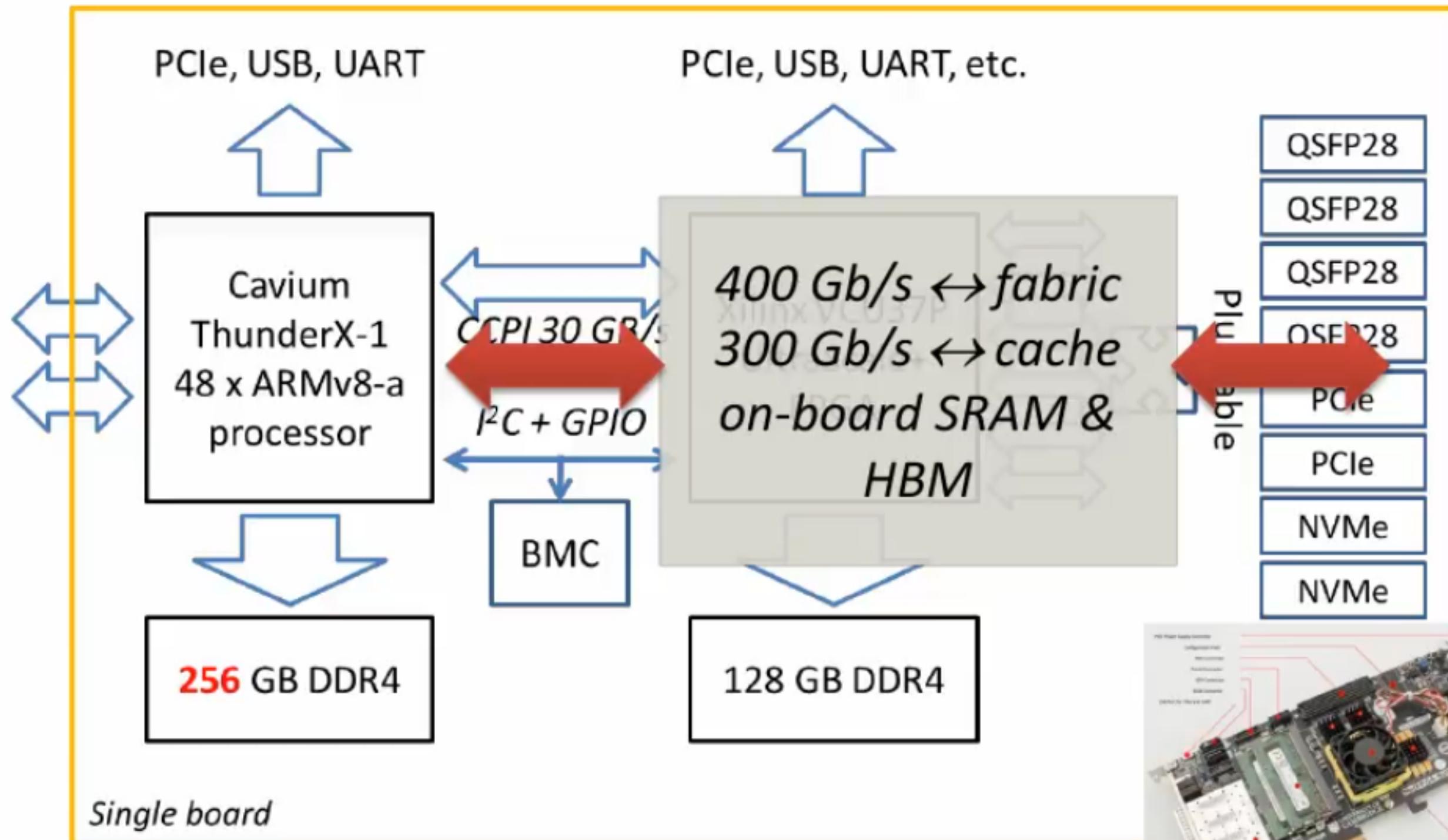
Systems@ETH Zürich





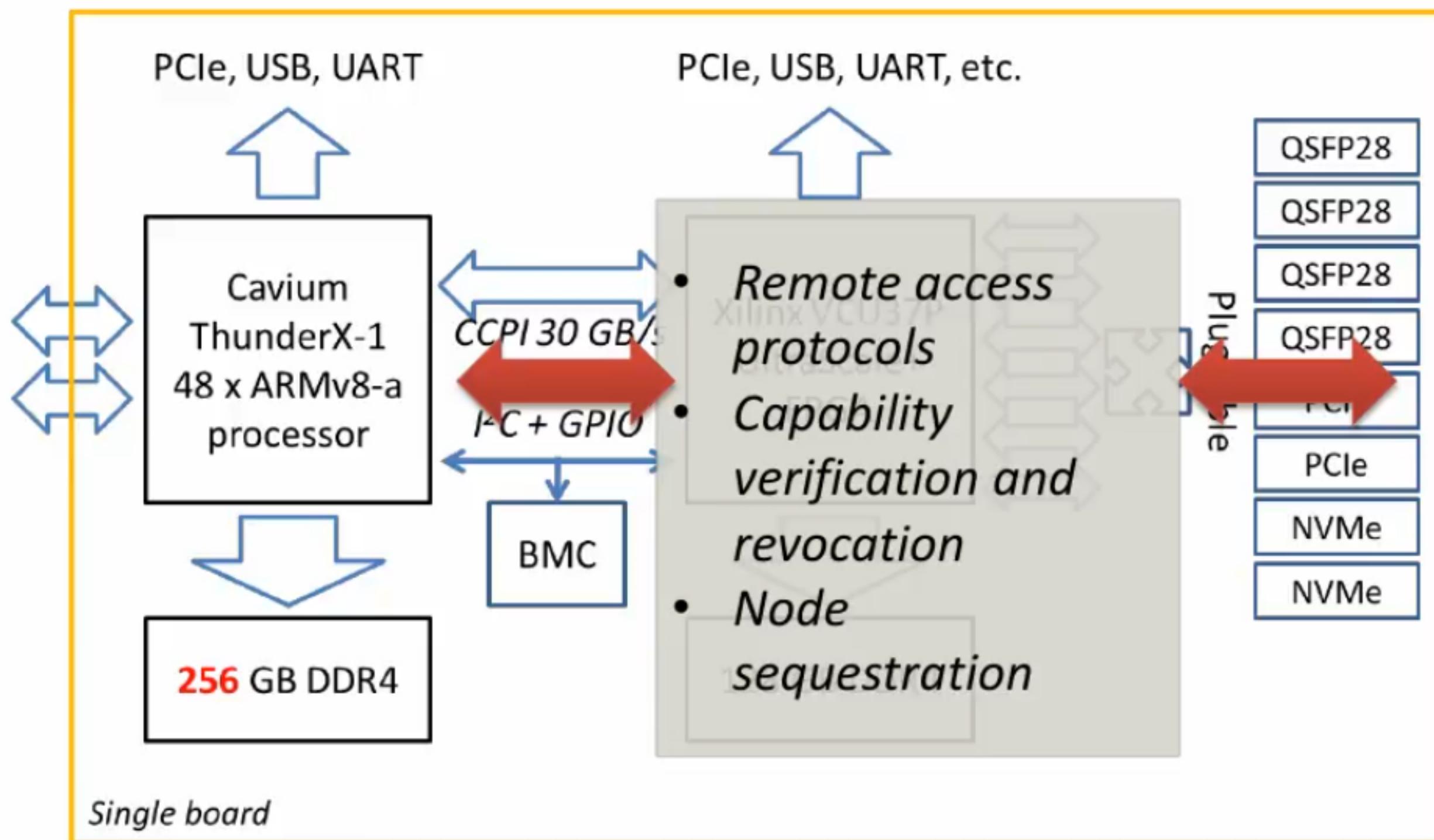
ENZIAN

The next NetFPGA?





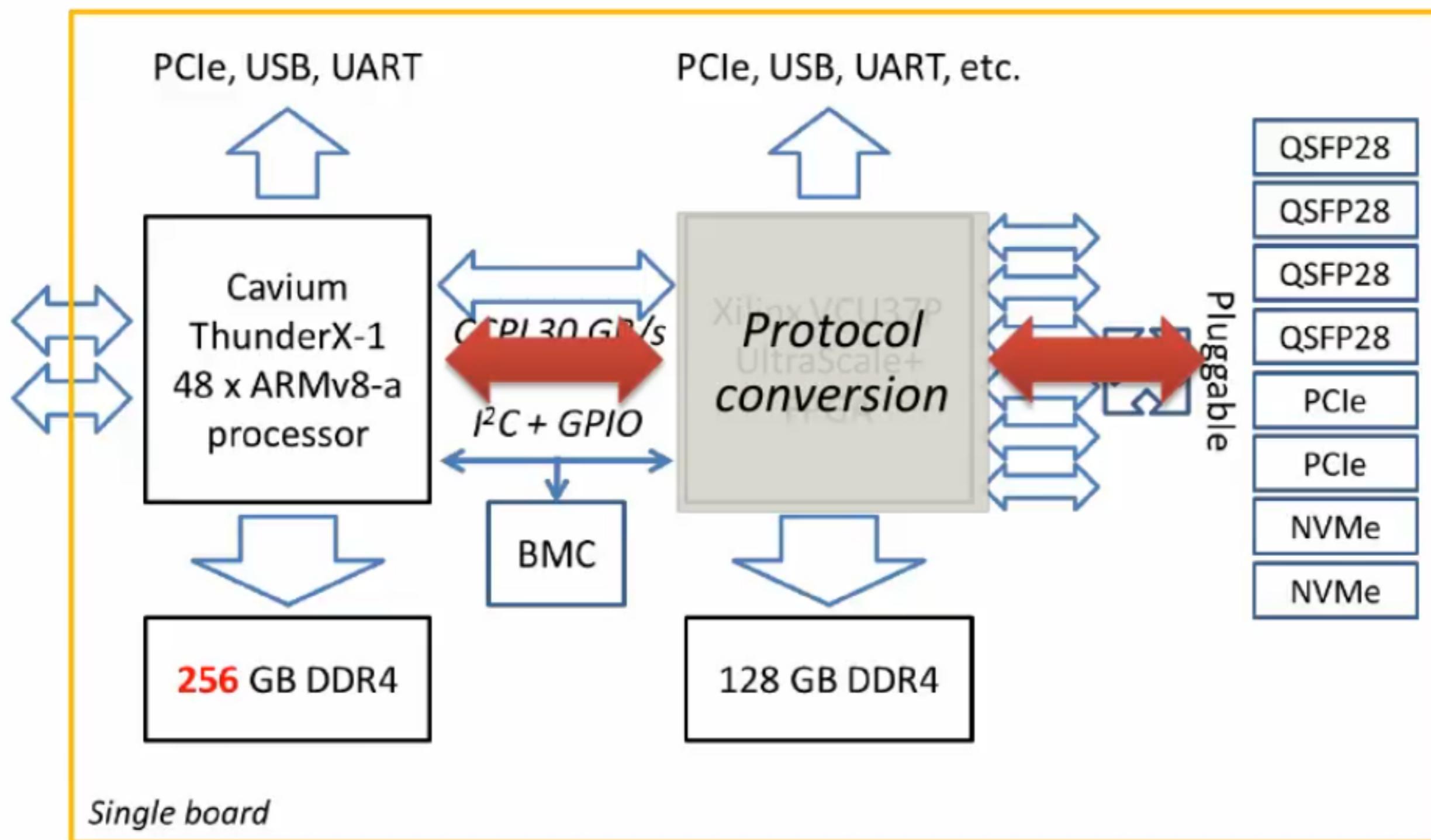
Remote memory





ENZIAN

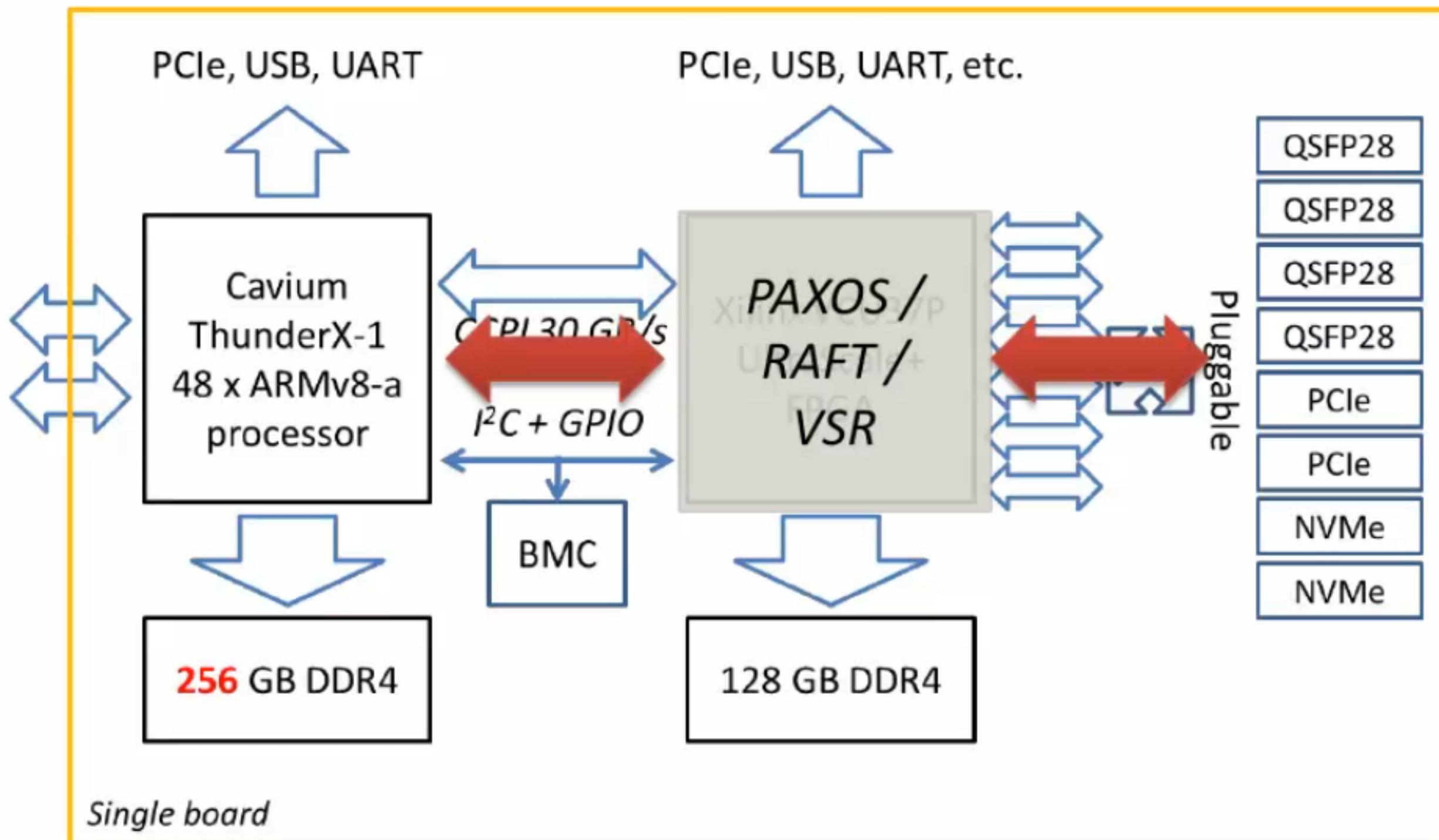
Extending coherency





ENZIAN

Offloading consensus



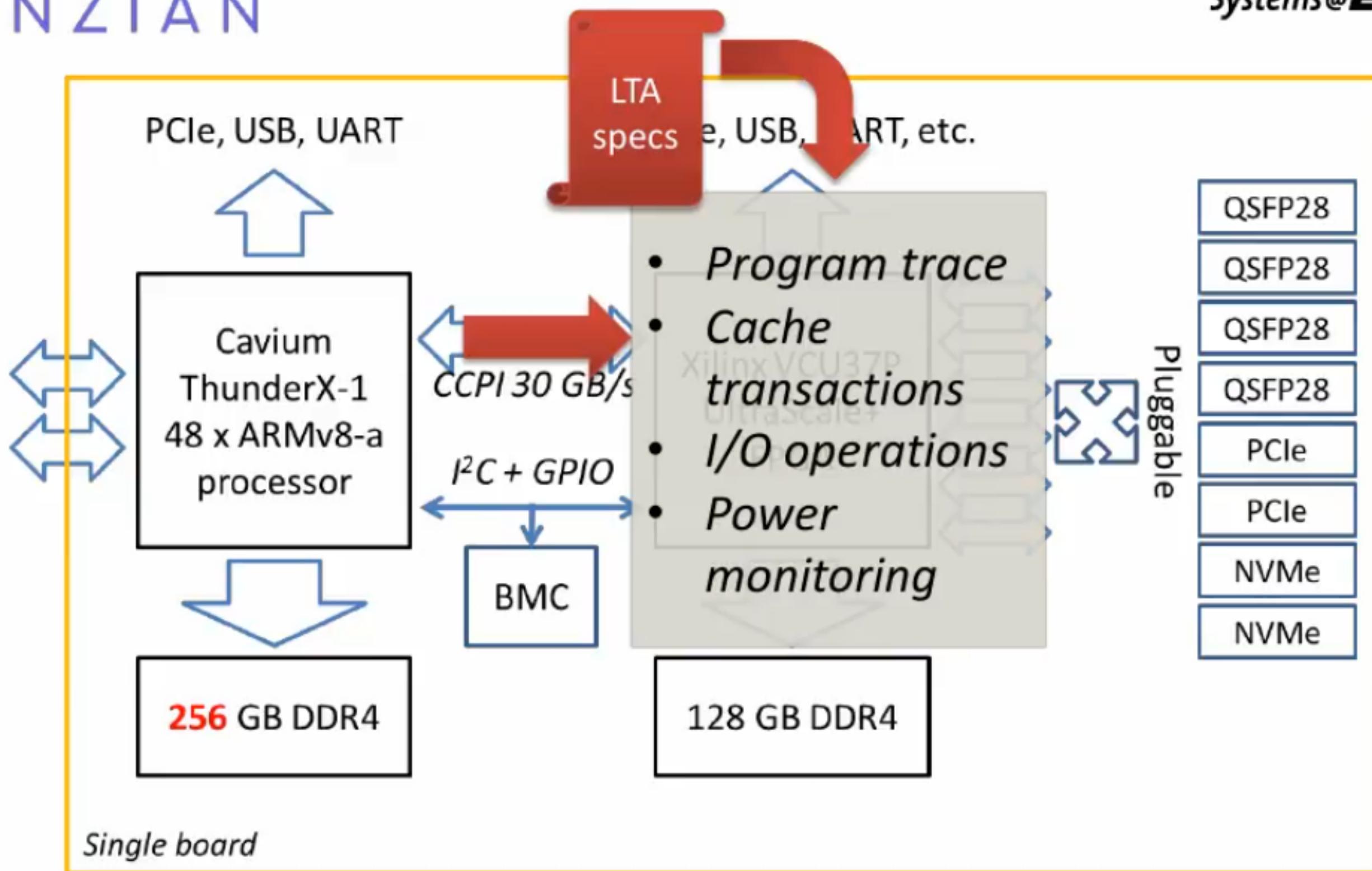


ENZIAN



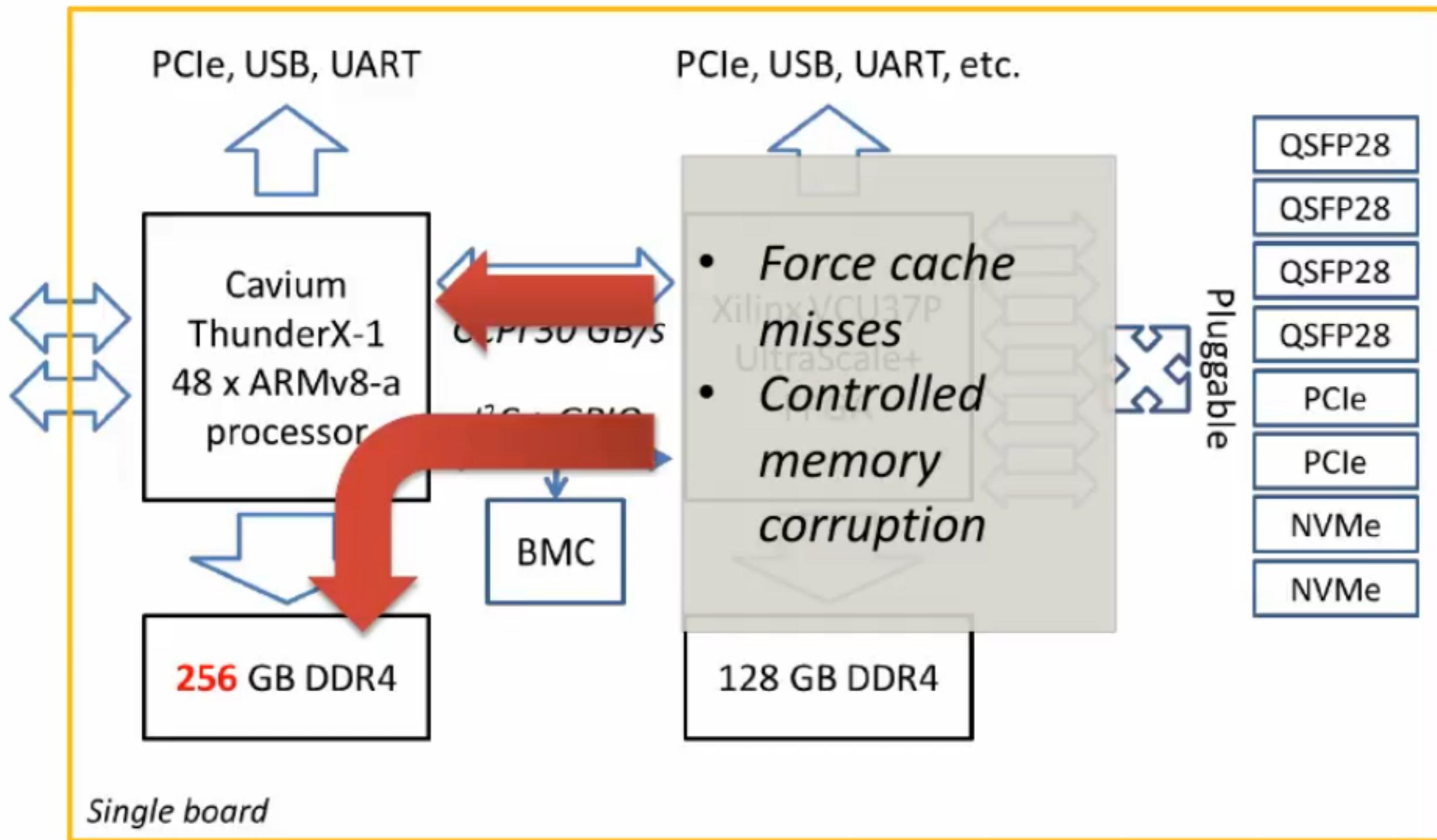
Systems@ETH Zürich

Runtime verification



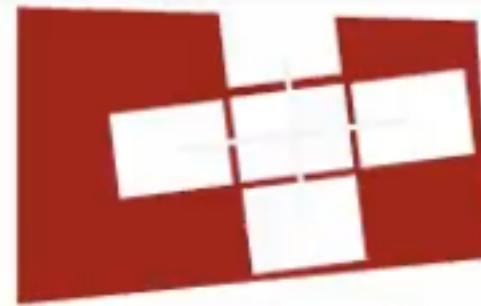


Fault injection



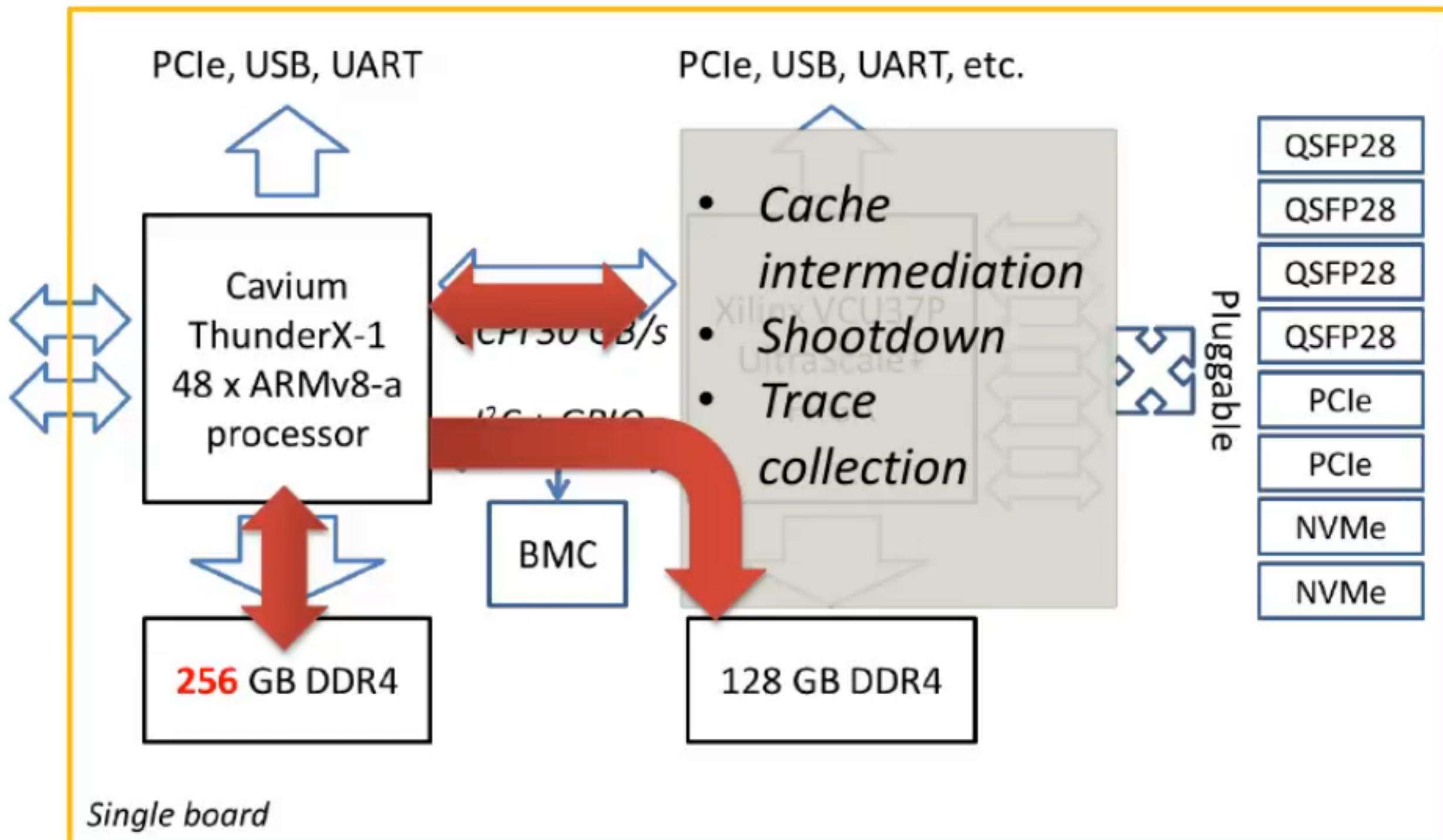


ENZIAN



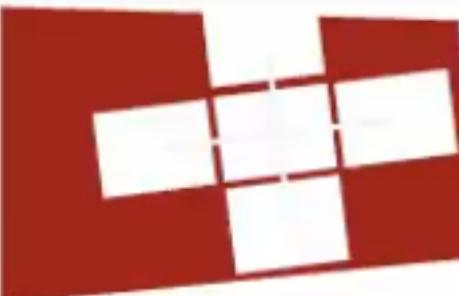
Systems@ETH Zürich

Cache profiling



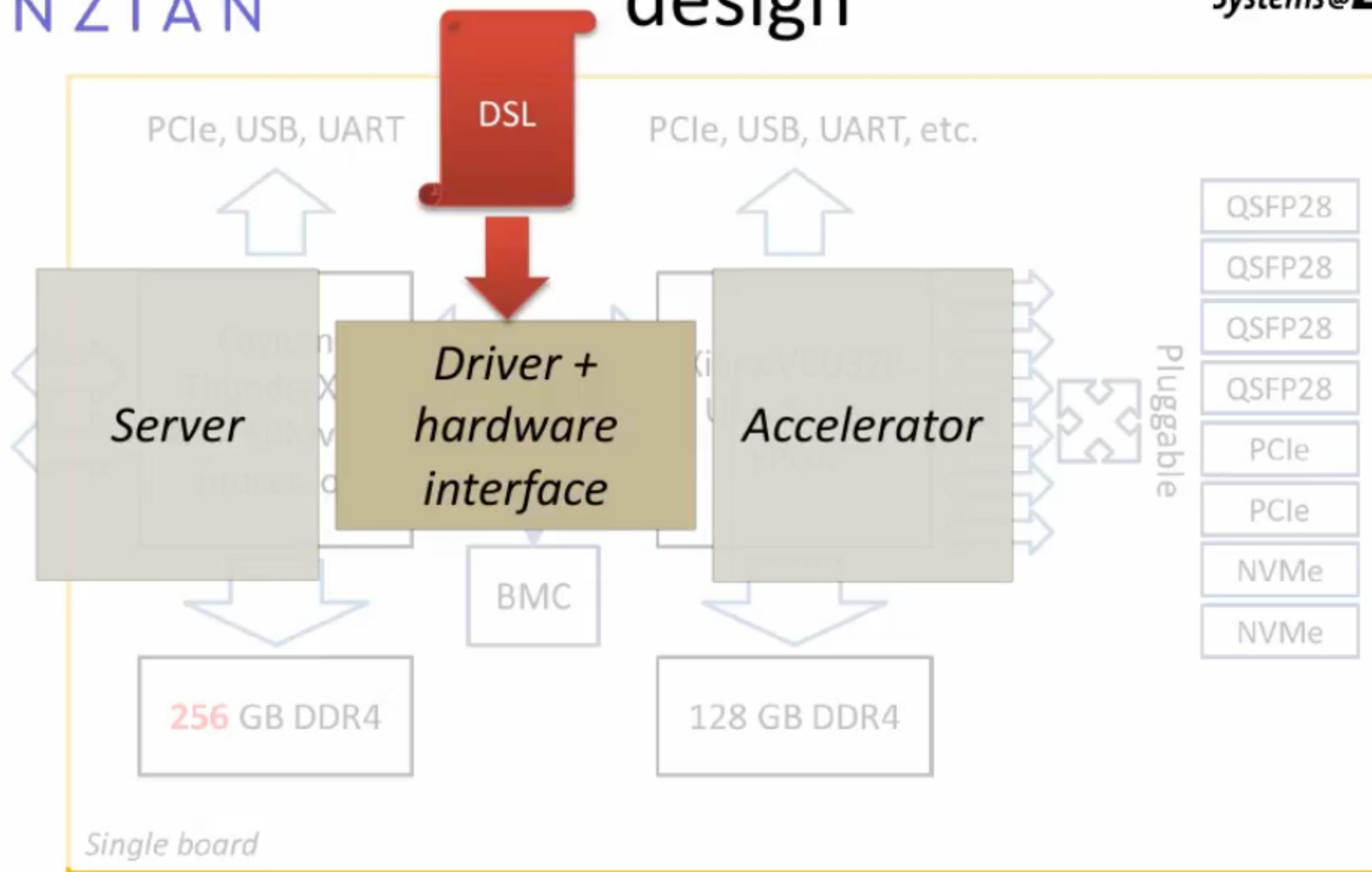


ENZIAN



Systems@ETH Zürich

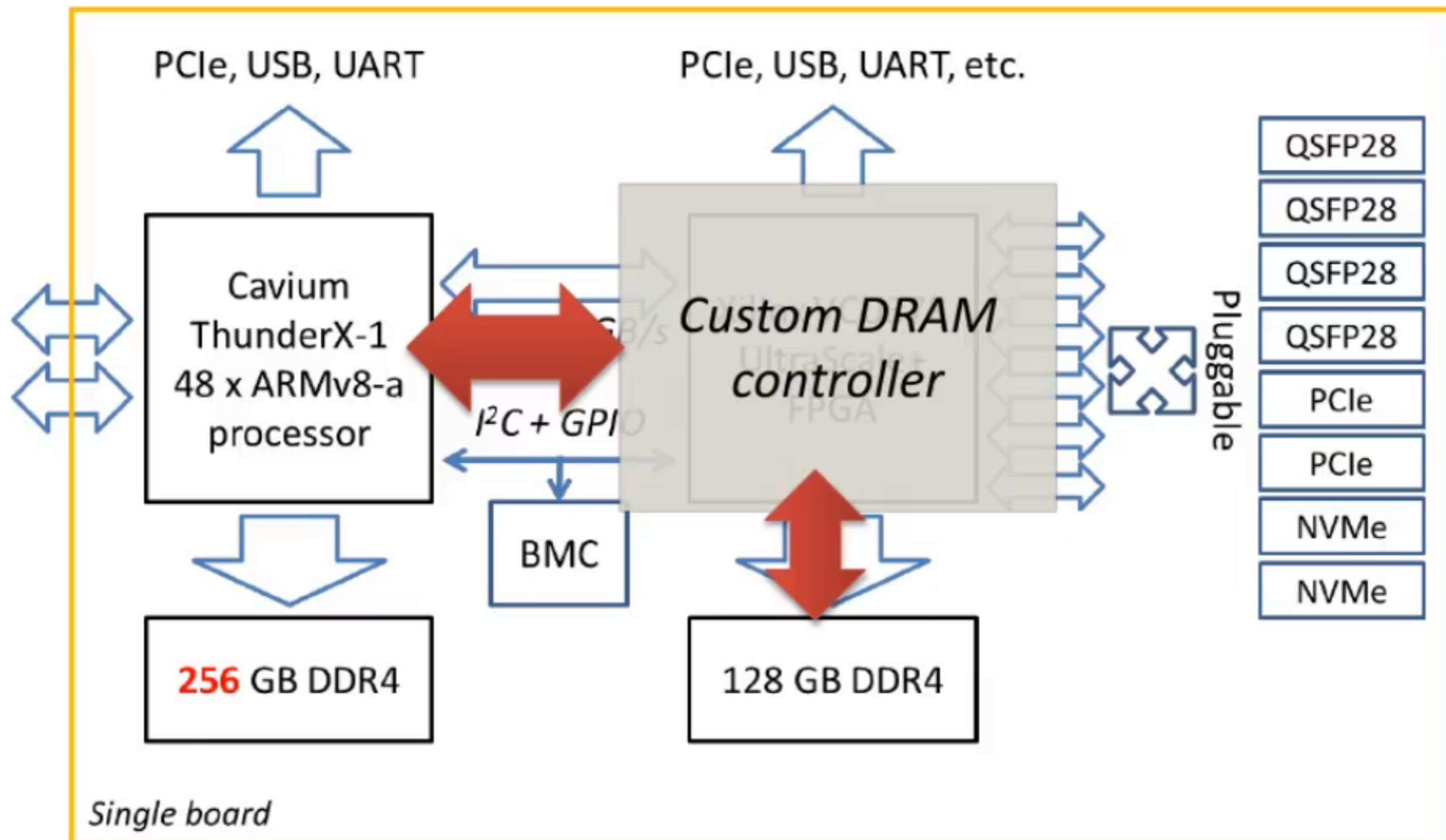
Server HW/SW co-design





ENZIAN

DRAM research



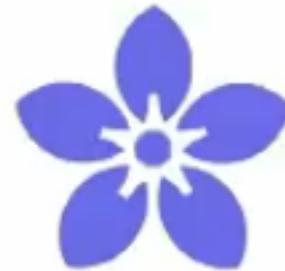
POSSIBLE IMPLICATIONS

Methodological questions

- ***Application*** use-cases, e.g:
 - NFV applications & control/data plane design
 - Database design exploiting hardware accelerators
- ***Systems*** use-cases, e.g.:
 - Accelerator interface design
 - OS support for closely-coupled FPGAs

Claiming validity

1. This **is** the platform
 - Why is this platform a valid design point?
 - How to meaningfully compare with others?
2. Emulating a **possible new** platform design
 - What would this mean if built from scratch?
 - What is the cost/benefit?



ENZIAN

Higher goal: research amplification



- Seed the research community
 - Remove major barrier to innovation in academia
- Precedents:
 - PlanetLab
 - Berkeley Unix
 - NetFPGA
 - Intel's DPDK
 - ...



DPDK

DATA PLANE DEVELOPMENT KIT



PLANETLAB

An open platform for developing, deploying, and accessing planetary-scale services

Conclusion

- COTS hardware is unrealistic
 - All the product action uses **custom** hardware
- Vendors can build almost **anything**
 - What to build is an **economic** question
⇒ not something we can answer
- We need **overengineered research platforms**
 - Our goal should be to deliver **options** and **techniques**



ENZIAN

www.enzian.systems