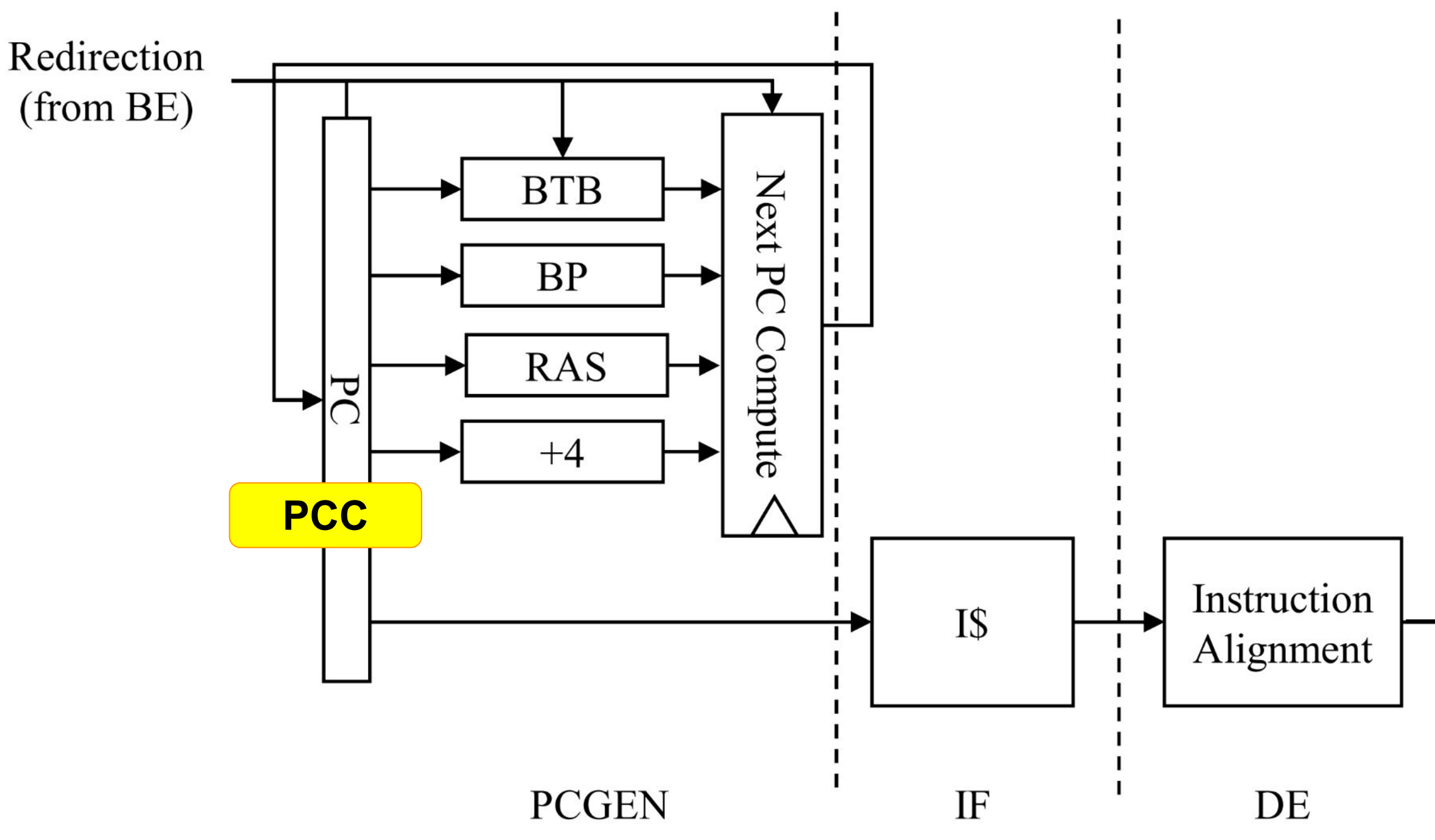
CHERI-Muntjac: an efficient, secure, application-class core

Yuecheng Wang, Jonathan Woodruff, Peter Rugg, Alexandre Joannou, Samuel W. Stark and Simon W. Moore
Department of Computer Science and Technology, University of Cambridge
yw737@cl.cam.ac.uk
Project URL: cheri-cpu.org

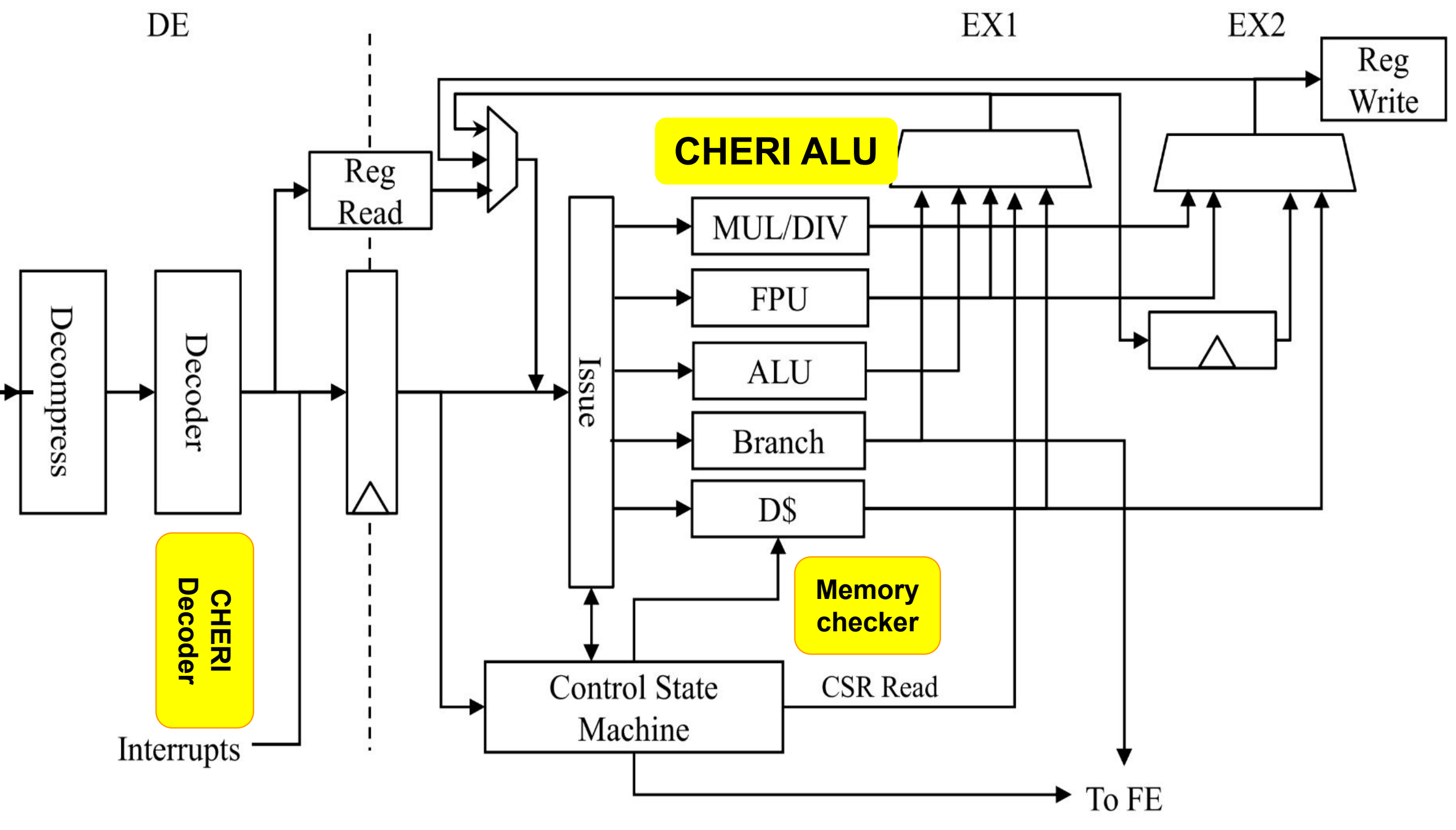
UNIVERSITY OF CAMBRIDGE
Computer Science & Technology

Muntjac is an open-source collection of components which can be used to build a multi-core, Linux-capable system-on-chip available from lowRisc. Muntjac has a few features that make it an interesting candidate for a CHERI extension. It has a simple 5-stage in-order pipeline which has small area usage and is easy to understand; nevertheless it implements the required extensions to boot Linux. Muntjac also has an interesting data cache implementation to support faster write-back and refill; it is written in SystemVerilog which is favored by industry, compared to Bluespec SystemVerilog, the language of most current open-sourced CHERI processors. In addition, Muntjac has a small area footprint on FPGA which allows evaluation of CHERI systems with higher concurrency than previously possible.

Muntjac frontend



Muntjac backend



Very small just like a muntjac

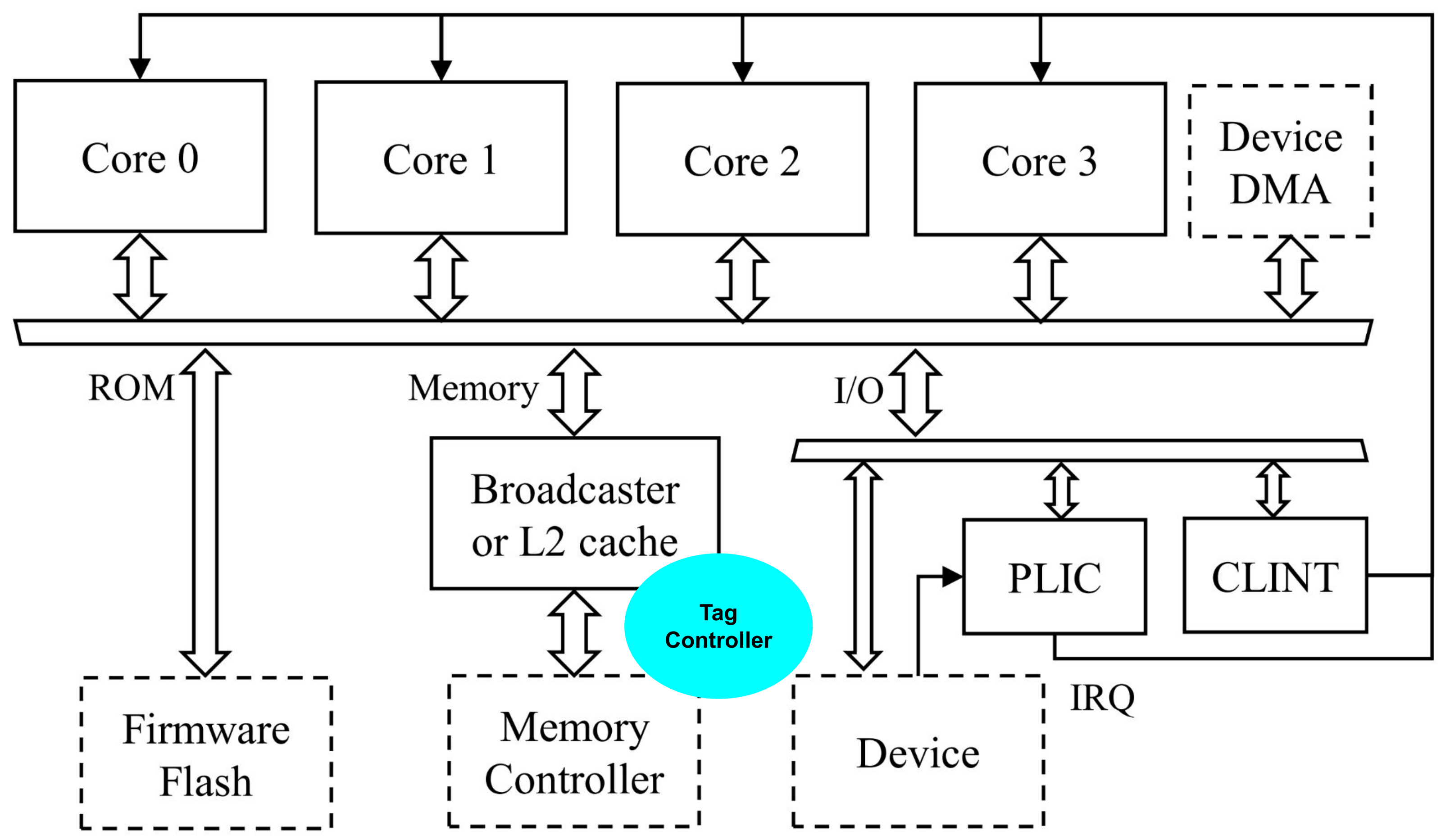
Similar to muntjac deer, our CHERI-Muntjac core is very small in area usage. One of our goals of CHERI-Muntjac is to be the smallest CheriBSD capable CHERI core, so that we can build a highly concurrent CHERI system.

Extending PC as PCC

The program counter (PC) is replaced with program counter capability (PCC), which limits control flow. Getting the PCC related behavior correct is the biggest challenge of extending CHERI to CHERI pipelines. Most of the development time has been spent on implementing PCC-related logic. TestRIG has been helpful to implement this correctly.

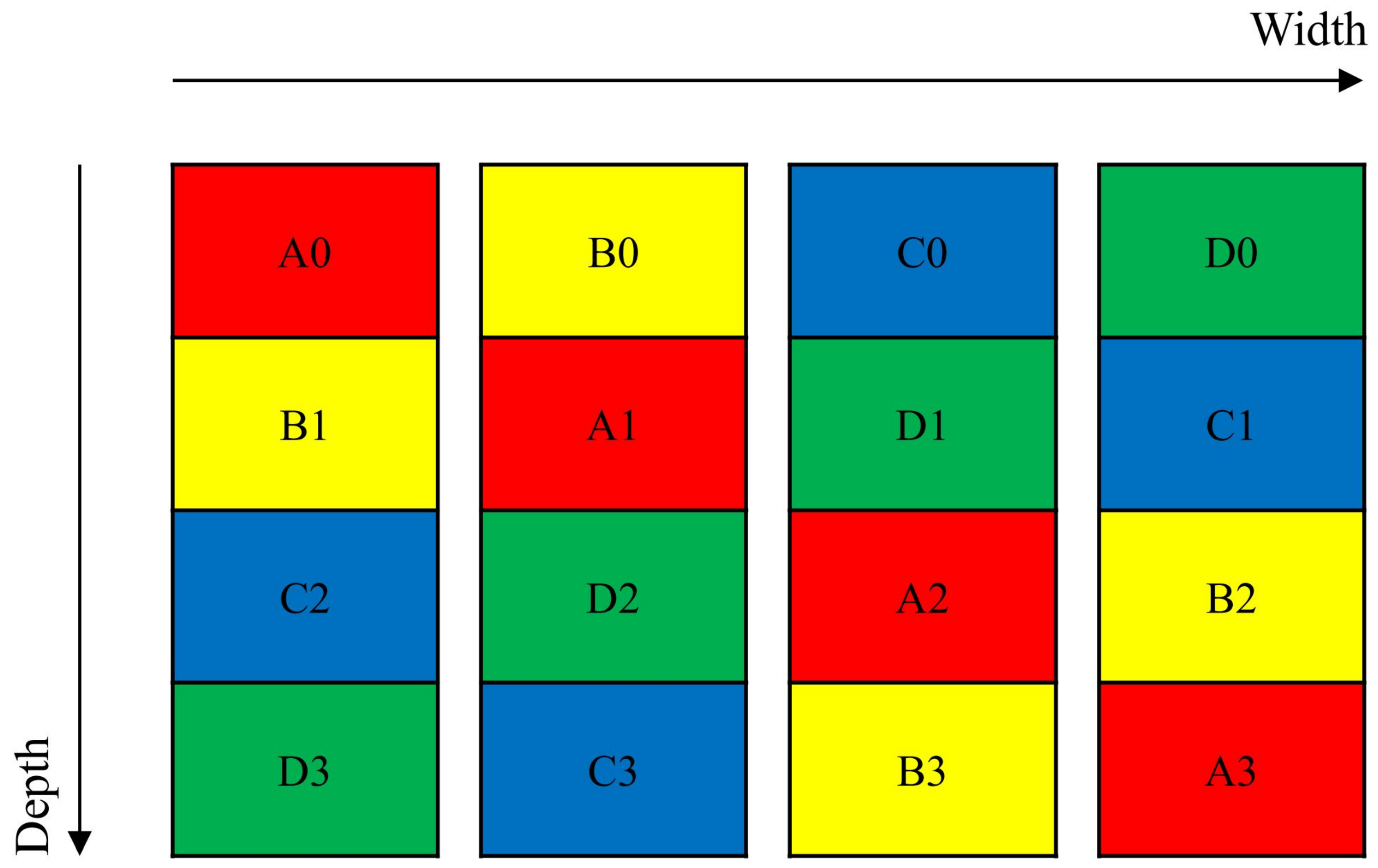
Changes made to backend

Besides the changes made to extend PC to PCC, the other major changes we made to the Muntjac pipeline are CHERI instruction decoding logic, a CHERI ALU for executing CHERI instructions and a memory checker to detect illegal access, e.g., dereferencing an untagged capability or out-of-bound access.



SystemVerilog Tag Controller

Our group has open-sourced a CHERI tag controller is written in BlueSpec SystemVerilog. We are implementing a CHERI tag controller written in SystemVerilog as part of the CHERI-Muntjac project. A simple version of this tag controller is functioning correctly now, and we plan to implement the hierarchical tag table optimization after CHERI-Muntjac boots CheriBSD. We also plan to use this design to evaluate several novel tag optimization strategies.



A0	B0	C0	D0
B1	A1	D1	C1
C2	D2	A2	B2
D3	C3	B3	A3

Efficient changes made to caches

Muntjac's cache system has a 64-bit interface to the pipeline which is supported by its 64-bit granule width SRAM. Instead of doubling the SRAM granule width to support 128-bit capability width accesses, we rely on data cache interleaving. As shown in the figure above, each block represents a 64-bit-wide data bank, and a single access can span multiple banks, which are also used in multiple ways. For instance, to write 128 bits of data, the data will be split into A0 and A1 and written separately; on read, we combine the output of A0 and A1 into one 128-bit word.

Verification

We have been using **TestRIG** to verify CHERI-Muntjac against CHERI-Sail to ensure implementation correctness TestRIG has been very useful to help implement CHERI correctly.
We are also using CHERIFreeRTOS in simulation to check its correctness.

Software bring-up

We are bringing up CheriFreeRTOS on CHERI-Muntjac in simulation, with the end goal of bringing CheriBSD up on CHERI-Muntjac on FPGA. After that, we will run CHERI revocation, SPEC and other benchmarks on Muntjac-CHERI to evaluate its scalability.

SRI International
CHERI