

April 2nd, 2025



CHERI

CVA6-CHERI Budding

Authors

Bruno Sá, **Jonathan Woodruff**, Peter Rugg, Alexandre Joannou, Sandro Pinto, Simon W. Moore

○ Goal of CVA6-CHERI Project

- Open-source reference design of to-be-ratified Zcheri RISC-V extension

- Accessible & useful to commercial ASIC designers
- Viable in FPGA or ASIC
- Seed experimentation with hardware and software prototypes

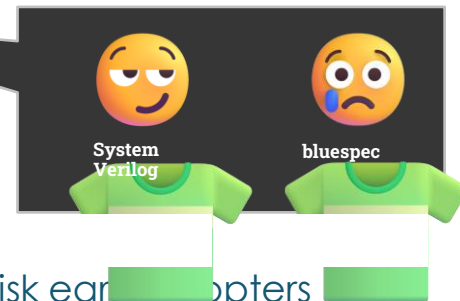
Reference design on ~£1000
Genesys 2 FPGA board

Library of reusable high-quality microarchitectural components
e.g. Tag controller, Compressed capability function library

- Ensure a second source for a compliant implementation to de-risk early adopters

- Start with a proven open-source design

- That has been shipped in silicon
- With an active open-source community



○ CVA6

A European Rocket
(those who know...)

Permissive license
soldierpad, from Apache 2.0

~54k lines of SystemVerilog
~760 modules

- Began life as Ariane at ETH Zurich for the PULP project ~2017
- Adopted as CVA6 at the OpenHW Group ~2020
- Extensively overhauled for modern, principled SystemVerilog style ~2022

👤 **Thales** for shouldering
CVA6 project coordination!

LowRISC style guide

- Very active community and development

~29 commits/month over
previous 3 months

~2019

Coherent
Multicore

~2019

Floating
Point

~2023

Vector

~2023

Hypervisor

~2024

Dual Issue

~2024

Coherent
Multicore II

○ CVA6-CHERI Project: A tale of two organizations



Zero-Day Labs



Goal: Initial, viable implementation to test interaction of hypervisor and CHERI for mixed-criticality, embedded systems

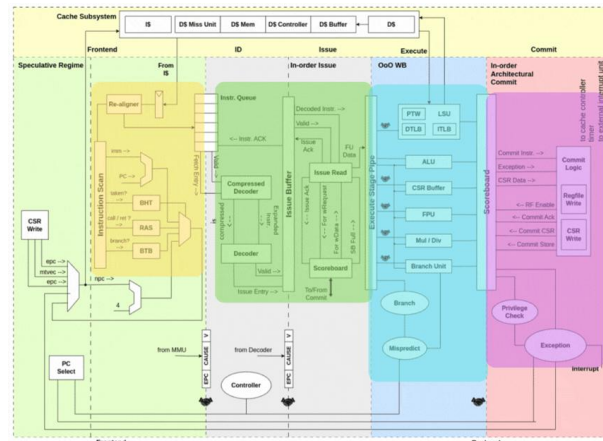
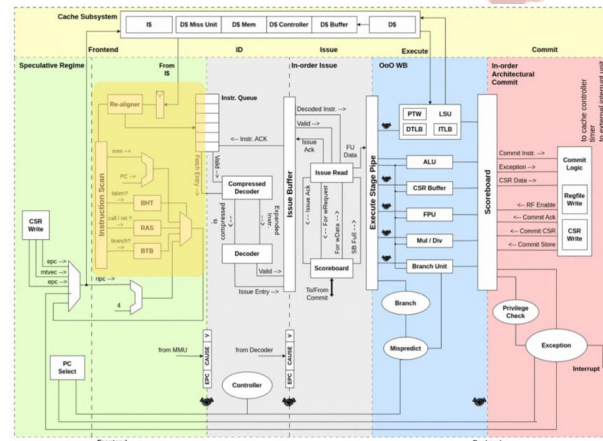
Capabilities Limited



Goal: Zcheri reference design of sufficient quality and performance to showcase the potential of CHERI-RISC-V implementations

TestRIG

- Existing CHERI-RISC-V verification uses TestRIG and the RVFI-DII interface
- Zero-Day Labs (Bruno Sá) developed an initial RVFI-DII interface
 - Used to bring up the experimental CHERI-RISC-V specification
 - One instruction at a time in the pipeline
- Capabilities Limited completed a robust implementation
 - Fully pipelined
 - Recovers from misprediction
 - Supports compressed instructions
- Zcheri support now in the TestRIG generator



SystemVerilog Capability Library



Reusable Zcheri microarchitectural functions in SystemVerilog and Bluespec

```
function Bool capInBounds(CapFat cap, TempFields tf, Bool inclusive);
// Check that the pointer of a capability is currently within the bounds
// of the capability
Bool ptrVStop = inclusive ? cap.addrBits <= cap.bounds.topBits
                           : cap.addrBits < cap.bounds.topBits;
// Top is ok if the pointer and top are in the same alignment region
// and the pointer is less than the top. If they are not in the same
// alignment region, it's ok if the top is in Hi and the bottom in Low.
Bool topOk = (tf.topHi == tf.addrHi) ? ptrVStop : tf.topHi;
Bool baseOk = (tf.baseHi == tf.addrHi) ? cap.addrBits >= cap.bounds.baseBits
                                       : tf.addrHi;
return topOk && baseOk;
endfunction
```

Bluespec

```
/**
 * @brief Function to check if capability is within bounds.
 * @param cap capability in register format.
 * @param cap_meta_data capability decoupled bounds meta data.
 * @param inclusive 0 - includes top in the in bounds check.
 *                  1 - excludes top in the inbounds check.
 * @returns 0 if not in bounds and 1 if in bounds.
 */
function automatic bool_t is_cap_reg_inbounds(cap_reg_t cap, cap_meta_data_t meta_data, bool_t inclusive)
    mw_t addr_mid = cap.addr_mid;
    bool_t check_addr = inclusive ? addr_mid <= cap.bounds.top_bits
                                : addr_mid < cap.bounds.top_bits;
    bool_t check_top = (meta_data.top_hi_r == meta_data.addr_hi_r) ? check_addr : meta_data.top_hi_r;
    bool_t check_base = (meta_data.base_hi_r == meta_data.addr_hi_r) ? addr_mid >= cap.bounds.base_bits
                                                                    : meta_data.addr_hi_r;
    return check_top && check_base;
endfunction
```

SystemVerilog

Tidy in both cases. Thanks to Bruno for developing a clean SystemVerilog style!

○ Capability Library Verification



- Use SymbiYosys-based equivalence proofs between Bluespec and SystemVerilog functions
- >21 functions proved equivalent
- Discovered ~5 bugs (one serious) in the initial library
- Used to guide port of library functions to Zcheri, which is nearly complete

```
`def_cap_mem_equiv_0arg(is_valid_reg, cap_mem_wrap_is_cap_reg_valid, module_wrap128_isValidCap, 1)
`def_cap_mem_equiv_1arg(set_valid_reg, cap_mem_wrap_set_cap_reg_valid, module_wrap128_setValidCap, 1, 129)
`def_cap_mem_equiv_0arg(is_valid, is_cap_mem_valid, module_wrap128_isValidCap, 1)
`def_cap_mem_equiv_1arg(set_valid, set_cap_mem_valid, module_wrap128_setValidCap, 1, 129)
`def_cap_mem_equiv_0arg(get_flags, get_cap_mem_flags, module_wrap128_getFlags, 1)
`def_cap_mem_equiv_1arg(set_flags, set_cap_mem_flags, module_wrap128_setFlags, 1, 129)
`def_cap_mem_equiv_0arg(get_hard_perms, get_cap_mem_reg_hperms, module_wrap128_getHardPerms, 12)
`def_cap_mem_equiv_0arg(get_addr, get_cap_mem_addr, module_wrap128_getAddr, 64)
`def_cap_mem_equiv_1arg(set_addr_unsafe, set_cap_mem_addr_unsafe, module_wrap128_setAddrUnsafe, 64, 129)
`def_cap_mem_equiv_1arg(set_addr_unsafe_reg, cap_mem_wrap_set_cap_reg_addr, module_wrap128_setAddrUnsafe, 64, 129)
`def_cap_mem_equiv_1arg(set_addr_inc_unsafe, set_cap_mem_addr_inc, module_wrap128_addAddrUnsafe, 11, 129)
`def_cap_mem_equiv_0arg(get_base, cap_mem_wrap_get_cap_reg_base, module_wrap128_getBase, 64)
`def_cap_mem_equiv_0arg(get_top, cap_mem_wrap_get_cap_reg_top, module_wrap128_getTop, 65)
`def_cap_mem_equiv_0arg(get_length, cap_mem_wrap_get_cap_reg_length, module_wrap128_getLength, 65)
`def_cap_mem_equiv_0arg(get_offset, cap_mem_wrap_get_cap_reg_offset, module_wrap128_getOffset, 64)
`def_cap_mem_equiv_1arg(set_addr, cap_mem_wrap_set_cap_reg_address, module_wrap128_setAddr, 64, 129)
`def_cap_mem_equiv_1arg(is_in_bounds, cap_mem_wrap_is_cap_reg_inbounds, module_wrap128_isInBounds, 1, 1)
`def_cap_mem_equiv_1arg(set_bounds, cap_mem_wrap_set_cap_reg_bounds, module_wrap128_setBounds, 64, 128)
`def_cap_mem_equiv_1arg_bsv_argconv(set_otype, cap_mem_wrap_set_cap_reg_otype, module_wrap128_setKind, otype_to_bsv_kind, 18, 129)
`def_cap_mem_equiv_1arg_bsv_argconv(seal, cap_mem_wrap_seal, module_wrap128_setKind, otype_to_bsv_kind, 18, 129)
`def_cap_mem_equiv_0arg(unseal, cap_mem_wrap_unseal, wrap_bsv_unseal, 129)
```

Macro which asserts
equivalence between
two functions

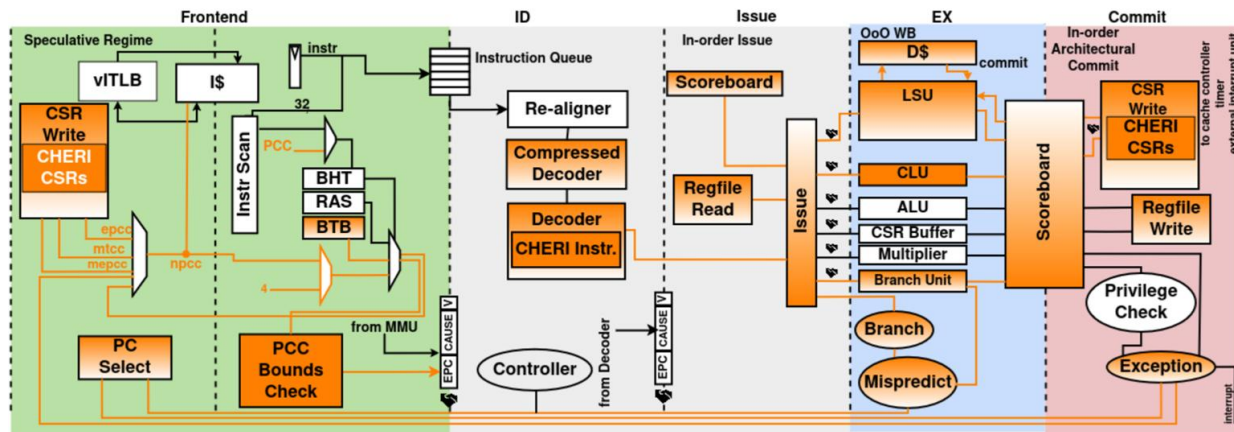
Name the two
functions to be
compared

Specify the size
of return type

○ Pipeline Changes for CVA6-CHERI

- Added new Capability Logic Unit (CLU)
- Widened register file & forwarding paths
- Widened PC to full capability & added bounds checks
- Added CHERI CSRs

What? Your processor doesn't have a CLU?



Bruno Sá, Donato Ferraro, José Martins, Everton Matos, Andrea Bastoni, and Sandro Pinto. "CHERI-RISC-V: A Case Study on the CVA6". In: *RISC-V Summit Europe* (June 2024).

○ Software Bringup on CVA6-CHERI

- We have booted **FreeBSD** on CVA6 Genesys2 FPGA (👏 Ruslan Bukin!)
- Bruno has booted **CheriBSD** on CVA6-CHERI
- We are adding networking support to the demonstration platform to improve usability

```
crypto: assign cryptosoft0 driver id 0, flags 0x000000
Device configuration finished.
procfs registered
Timecounters tick every 1.000 msec
lo0: bpf attached
vlan: initialized, using hash tables with chaining
IPsec: Initialized Security Association Processing.
md0: Embedded image 83886080 bytes at 0xfffffc00be29b0
tcp_init: net.inet.tcp.tcbhashsize auto tuned to 8192
random: unblocking device.
Trying to mount root from ufs:/dev/md0 [...].
WARNING: WITNESS option enabled, expect reduced performance.
regulator: shutting down unused regulators
Warning: no time-of-day clock registered, system time will not be set accurately
start_init: trying /sbin/init
Setting up sysctls
sysctl: unknown old 'kern.polling.user_frac' at line 2
sysctl: unknown old 'machdep.unaligned_log_pps_linit' at line 5
kern.coredump: 1 -> 0
mount / rw
entropy read from /boot/entropy
entropy read from /var/db/entropy/entropy.0
entropy read from /var/db/entropy/entropy.1
create 500m TMPFS at /tmp
set up loopback
lo0: link state changed to UP
generate host keys
start sshd
Logging in as root...
2025-03-20T01:48:13.336279+00:00 - login 40 - - login on console as root
# cd ..
# cd bin
# ./cheribsdtest-purecap

usage:
  cheribsdtest-purecap [options] -l          -- List tests
  cheribsdtest-purecap [options] -a          -- Run all tests
  cheribsdtest-purecap [options] <tests> [...] -- Run specified tests
  cheribsdtest-purecap [options] -g <glob> [...] -- Run matching tests

options:
  -f -- Only include "Fast" tests
  -c -- Enable core dumps
  -d -- Attach debugger before running test
  -s -- Sleep one second after each test
  -q -- Enable qemu tracing in test process
  -Q -- Enable qemu tracing in test process (user-mode only)
  -v -- Increase verbosity
  -x -- Output JUnit XML format

# ./cheribsdtest-purecap -a
TEST: bounds_global_static_uint8: Check bounds on global static uint8_t
PASS: bounds_global_static_uint8
```

CheriBSD boot on CVA6-CHERI on FPGA



○ Pre-initial Area numbers

- Bruno reported very early area number in June 2024
- We have discovered multiple opportunities for future improvement
- Zcheri will simplify the ISA
- Overhead will only decrease!

Single core CVA6H-CHERI with TagController support targeting Genesys2 FPGA (no optimizations yet)

Module	FPGA Resources (+ % overhead compared to vanilla CVA6)	
	LUTs	FFs
CVA6 SoC	101587 (+36%)	56107 (+33%)
CVA6 Core	68991 (+38%)	26284 (+31%)
wt_cache_subsystem	9963 (+23%)	3034 (+36%)
load/store unit	13818 (+12%)	9010 (+13%)
execute stage	26752 (+16%)	11756 (+12%)
cheri logic unit	1953	0
issue_stage	22366 (+72%)	4656 (+38%)
AXI Tag Controller	7948	7762

Bruno Sá, Donato Ferraro, José Martins, Everton Matos, Andrea Bastoni, and Sandro Pinto. "CHERI-RISC-V: A Case Study on the CVA6". In: *RISC-V Summit Europe* (June 2024).



CHERI

THANK YOU

Q & A

