

ORCONF 2023 Munich, 2023-09-15 Michael Gielda, mgielda@antmicro.com



ANTMICRO

- Founded in 2009, dedicated to commercial adoption of open source in the hardware space, Platinum CHIPS Alliance & Founding RISC-V member
- Long term FOSSi supporter
- Engineering services, software/hardware co-design, tools and R&D
- Developing, adapting and integrating open source ASIC and FPGA tooling, IP, software, hardware and AI
- Help customers achieve vertical integration, address niche use cases, increase productivity





CALIPTRA

- Open source integrated Root of Trust block
- Collaboration between Google, Nvidia, Microsoft and AMD within CHIPS Alliance (spec lives in OCP, implementation is developed in CHIPS)
- Focused on specific set of features
- By default meant to be integrated with a larger SoC, with concrete plans from several member companies
 - has both Boot Media Integrated/Dependent variants
- Uses VeeR EL2 CPU core
- Reuses a number of OpenTitan peripheral cores

E Caliptra



CALIPTRA - STATUS

- Moving fast (and trying not to break things)
- Going towards 1.0 RTL release as planned
- Focusing on a small but useful first version, without too many extra features
- Slated to be included in multiple chips coming from main partners (some time from now)

년 Caliptra

antmicro



CHIPS MEMBERS INVOLVED WITH CALIPTRA WG







CALIPTRA AND ANTMICRO

- Focus on enabling open source collaboration on the project for both current project partners and external adoption
- Took over cleanup and maintenance of VeeR cores family, primarily EL2, make sure it's on par
- Another goal enable more open source collaboration around RTL with open source tools:
 - Developer productivity
 - System-level testing and integration
 - Public-facing Cl



- E. Caliptra





VEER

- Family of open source 32 bit RISC-V CPU cores
- Three options
 - EL2 tiny and low power core this is the one used in Caliptra
 - EH1 high performance core
 - EH2 dual threaded successor to EH1
- All the variants are implemented with SystemVerilog
- RTL is ASIC proven, shipped in millions of devices



VERIBLE

- SystemVerilog linter, formatter, indexer etc. developed by Google and donated to CHIPS Alliance
- A practical tool used extensively in the wild!
- Now also in Caliptra



VERIBLE GH ACTIONS

- Linter and formatter actions
 - chipsalliance/verible-formatter-action
 - chipsalliance/verible-linter-action
- Both linter and formatter enabled for VeeR EL2
 repository

desi		mem.sv				
				mport el2_pkg		
103	103	.clk_	_override(dccm_clk_	override)		
105	105					
106			else begin: Gen_dcc	_disable		
	106					
		+ begin: G	en_dccm_disable			
0	github	actions	ot)4 hours ago			
⚠ [verible-verilog-lint] review by revlewdog ♥ All generate block labels must start with g_ or gen_[Style: generate-constructs] [generate-label-prefix]						
desi	ign/el2_	mem.sv				
	118					
0	 github-actions (bot) 4 hours ago [verible-verilog-lint] review by reviewdog@ All generate block statements must have a label [Style: generate-statements (generate-label] 					
	⚠ [ver All gene [genera	ible-verilo erate block ate-label]	g-lint] review by statements must	reviewdog🤣 have a label [S	tyle: generate-statement	5]
desi	▲ [ver All genera [genera	ible-verilog erate block hte-label] veer_wrapp	g-lint] review by statements must ner.sv	reviewdog 🤣 have a label [S	tyle: generate-statement	5]
desi	All genera [genera ign/el2_ 305	ible-verilo erate block ite-label] veer_wrapp + output	g-lint] review by statements must er.sv t logic	reviewdog 🐼 have a label [S dec_	tyle: generate-statement tlw_perfcnt1,	5]
des 3	[ver All genera [genera ign/el2_ 305 306	ible-verilog erate block ate-label] veer_wrapp + output + output	g-lint] review by statements must er.sv t logic t logic	reviewdog® have a label (S dec_ dec_ dec_	tyle: generate-statement _tlu_perfcnt1, _tlu_perfcnt2,	5]
desi 305	▲ [ver All gene [genera ign/el2_ 305 306 github-	ible-verilog erate block ite-label] veer_wrapp + output + output eactions	g-lint] review by statements must er.sv t logic t logic ot) 4 hours ago	reviewdog® have a label [S dec_ dec_	tyle: generate-statement _tlu_perfcnt1, _tlu_perfcnt2,	5]
desi 305	A [ver All gene [genera 305 306 github- Remove	ible-verilog erate block te-label] + output + output + output + output ible-verilog e trailing sp	g-lint] review by statements must er.sv t logic t logic ot) 4 hours ago g-lint] review by paces. [Style: trailin	reviewdog © have a label [S dec. dec. reviewdog © reviewdog ©	tyle: generate-statement tlu_perfcnt1, _tlu_perfcnt2, trailing-spaces]	5]
des 1 305	▲ [ver All gene [general ign/el2_ 305 306 github- ▲ [ver Remove Sugges	ible-verilog erate block ite-label] + output + output + output eractions b ible-verilog t trailing sp	g-lint] review by statements must er.sv t logic t logic ot) 4 hours ago g-lint] review by paces. [Style: trailin	reviewdog © have a label [S dec_ dec_ reviewdog © ig-spaces] [no-	tyle: generate-statement _tlu_perfont1, _tlu_perfont2, trailing-spaces]	5]
des 1 305	▲ [ver All gene [general ign/el2_ 305 306 github- ▲ [ver Remove Sugges 117	ible-verilo; erate block ite-label] veer_wrapp + output + output eactions b ible-verilo; e trailing sp ited changes 305	g-lint] review by statements must er.sv t logic t logic ot) 4 hours ago g-lint] review by paces. [Style: trailin	reviewdog © have a label [S dec_ dec_ neviewdog © ng-spaces] [no-	tyle: generate-statement _tlu_perfcnt1, _tlu_perfcnt2, trailing-spaces] 	5]
des 1 305 Ç	<pre></pre>	ible-verilo; erate block ite-label] veer_wrapp + output + output + output etrailing sp ited changes 305 - 117 -	g-lint] review by statements must er.sv t logic t logic ot 4 hours ago g-lint] review by paces. [Style: trailin output logic output logic	reviewdog © have a label [S dec. dec. g-spaces] [no-	tyle: generate-statement	5]
des 3 305 ()	▲ [ver All gene [general ign/el2_ 305 306 github- ▲ [ver Remove Sugges 117	ible-verilo; veer_wrapp + output + output + output actions b ible-verilo; trailing sp sted changes 385 - 1 177 - 118 +	g-lint] review by statements must wer.sv t logic t logic ot 4 hours ago g-lint] review by baces. [Style: trailin output logic output logic output logic	reviewdog 👁 have a label (S dec. dec gespaces) [no-	tyle: generate-statement _tlu_perfcnt1, _tlu_perfcnt2, trailing-spaces] dec_tlu_perfcnt1, dec_tlu_perfcnt1, dec_tlu_perfcnt1,	5]

 \mathbf{O}

VEER TESTING

- The open source release of VeeR cores family originally only included system level tests
 - A more complex test suite was internal to the original authors and was not released as open source
- One of the main goals is to extend VeeR's publicly available test suite with tests for specific core components
- We added public CI to the VeeR reportunning those tests for every commit
- The suite includes code coverage reporting the Cl generates a summary webpage



VEER TESTING

- Collecting line and toggle coverage from several types of tests:
 - RISCV-DV
 - RISCOF (RISC-V Architectural Test Framework)
 - UVM (for specific VeeR blocks, i.e. PIC, DEC, EXU, IFU, LSU, etc.)
- Coverage results built within a continuous integration workflow for main branch and PRs

. Tre Chips Alliance

Current view: top level - design/lsu Test: all Date: 2023-09-15 12:54:43

	HIL	Total	Coverage
Lines:	184	206	89.3 %

Filename	Branch Coverage 🗢	
el2_lsu_bus_buffer.sv	86.0 %	135 / 157
el2_lsu_bus_intf.sv	100.0 %	6/6
el2_lsu_lsc_ctl.sv	100.0 %	17/17
el2_lsu_stbuf.sv	100.0 %	22 / 22
el2_lsu_trigger.sv	100.0 %	4/4

Generated by: LCOV version 1.16

PHYSICAL MEMORY PROTECTION (PMP)

- Originally, in VeeR-EL2 Core only a rudimentary PMP was available. Currently, a RISC-V compliant PMP is being developed.
- Physical Memory Protection is needed to support secure processing and contain faults (PMP violations are precisely trapped at the processor)
- The PMP limits the physical addresses accessible by software:
 - Divide the memory into regions with different levels of privilege
 - Manage protection rules (Read, Write, Execute)
 - Grant or revoke permissions to Supervisor, User and Machine mode



JTAG INTEGRATION

- For debugging the core, Caliptra added a JTAG interface, which needed its own end-to-end tests
 - The test connects to a simulated SoC with
 OpenOCD and runs typical debug scenarios
 - Run in public CI on GitHub with open source tools
- Helped uncover a bug earlier this year which is now covered by a dedicated test



OPEN SOURCE VERIFICATION

- Caliptra's primary use case is integration into bigger SoCs
- The primary testing methodology is UVM based based on widespread use in founding organizations
- UVM simulations are not yet possible with open source tools only
- There is an ongoing effort on enabling UVM simulation in Verilator (see Chris' presentation tomorrow)
- To enable open source collaboration around improving testing coverage, we have been using Cocotb and pyuvm testing which can be run completely open source in CI









RISCV-DV

- riscv-dv open source instruction generator for RISC-V processor verification
- By default uses UVM, but also features a Python generator
- To make it work with the current EL2 codebase, VeeR's out of order div/rem block required special handling in the generator
 - We improved trace analyzer and code generator to handle the out-of-order division logic
- fixed support in the Python-based generator, riscv-dv tests are now part of VeeR's public CI
- Also using the VeeR tests in RISC-V DV CI!

addi sp, sp, 16	01010113	0x8000bfa8:
ret	00008067	0x8000bfac:
li a5, -1	fff00793	0x8000c0b8:
beq a0, a5, 28	00f50e63	0x8000c0bc:
bge s1, a0, 24	00a4dc63	0x8000c0c0:
andi s0, s0, 8	00847413	0x8000c0d8:
csrrs s0, mstatus, s0	30042473	0x8000c0dc:
lw ra, 28(sp)	01c12083	0x8000c0e0:
lw s0, 24(sp)	01812403	0x8000c0e4:
lw s1, 20(sp)	01412483	0x8000c0e8:
addi sp, sp, 32	02010113	0x8000c0ec:
ret	00008067	0x8000c0f0:
jal -41504	de1f50ef	0x8000cac4:
li a5, 8	00800793	0x800028a4:
csrrs a5, mstatus, a5	3007a7f3	0x800028a8:
wfi	10500073	0x800028ac:
addi sp. sp80	fb010113	0x80000010:

lantmicro

CALIPTRA FIRMWARE VALIDATION

- Rust-based rv32imc interpreter with peripherals
 - can run all integration tests
 - all new firmware tested pre-merge in CI
- Verilator tests
 - direct from RTL but (expectedly) much slower
 - minutes to compile, hours to run
 - used for several important nightly tests
 - allow outputting signal data into .vcd files for analysis in e.g. GTKwave
- Good test infrastructure
 - test case integration with VS code

\equiv	EXPLORER	Ibrs Smoke_test.rs X		
	✓ CALIPTRA-SW	test > tests > @ smoke_test.rs > @ retrieve_csr_test		
р Р	> target • .gitignore © Cargo.lock • LICENSE	16 fn assert_output_contains(haystack: &str, needle: &str) (17 assert(1 18 haystack.contains(needle), 19 "Expected substring in output not found: (needle)"	A sector of the	
2	README.md systemrdl	20); 21) 22 affrest	Martin States and American	
₽	> target > test > src > tests	File Teu(Debu) 4 fn tettieve_str_test() { cost GENERATE_DDEVID_CSR: u32 = 1; cost GENERATE_DDEVID_CSR: u32 = 1;		
	> smoke_testdata smoke_test.rs Cargo.toml	21 It for the solid product and solid (index) to (above, a memory), whereas (interpreted) is a solid product and the solid produc		
	 README.md test-harness ureg x509 	31 Orfault:default() 32). 33 initial.dbg.manf.service.reg: GENERATE_IDEVID_CSR, 34 Orfault:dbg.manf.service.reg: GENERATE_IDEVID_CSR,	The second secon	
	.gitignore .gitmodules Cargo.lock Cargo.tock	<pre>35 1) 36umrap(); 37 38 let mut txn = hw.wait_for_mailbox_receive().umwrap(); 39</pre>		
	> OUTLINE > TIMELINE > RUST DEPENDENCIES	<pre>39 let csr_der = mem:take(&mut <u>twn.req.data</u>); 40 <u>twn.respond_success();</u> 41 let csr = newsel:vs00:vs00:vs00:errom der(&csr_der) unwran(); 42 let csr = newsel:vs00:vs00:vs00:errom der(&csr_der) unwran();</pre>		
	> aes 0.8.2 > alloc	<pre>42 let CSL = openSitionSitionSitionSite(cst_open(sca_pen))onmap(); 43 let Cst_tkt = Siting:from_utfS(cst.optext(),umerap()).umerap(); 44 // To update the CSR testdata;</pre>		
8	> anyhow 1.0.70 > arrayref 0.3.7 > asn1 0.13.0	<pre>46 // std::fs::write("tests/smoke_testdata/idevid_csr.txt", &csr_txt).umwrap(); 47 // std::fs::write("tests/smoke_testdata/idevid_csr.der", &csr_der).umwrap(); 48</pre>		
-	> asn1_derive 0.13.0 > atty 0.2.14	<pre>49 printla("csr: ()", csr_txt); 50 c1 ascent en1/</pre>		

CALIPTRA FIRMWARE VALIDATION

- FPGA emulation
 - Uses zcu104 (UltraScale+)
 - Tests run on Cortex-A53 Linux
 - Both fast and accurate, but cannot (yet) run all the tests
 - Long compile time



RTL2GDS FLOW WITH OPENROAD

- By default Caliptra uses proprietary tools for the digital design, as do its main members
- They are all SystemVerilog shops
- To enable fast turnaround feedback, we have an ongoing effort on getting open source
 VeeR/Caliptra RTL2GDS flow directly from SV source using Antmicro's SystemVerilog plugin for Yosys through Surelog/UHDM in OpenROAD

OpenROAD



CUSTOM GH RUNNERS

- Since we need more resources than the standard GH runners can provide (long verification jobs etc) as well as we need to allow use of proprietary tools in CI, we're using Antmicro's open source custom GH runners
- Work to structure results between public artifacts and ones we (sadly) cannot share publicly, available to project members
- Infrastructure is open source and available on GitHub, used by a number of CHIPS projects



CUSTOM GH RUNNERS - EXECUTION METRICS

- Each job running on our custom runners outputs a performance plot
- Helps developers to better understand how to optimize their builds
- Uses our open source Sargraph tool



LEARN MORE

- OCP Summit
 - San Jose, October 17-19
 - Register at opencompute.org



DCP GLOBAL SUMMIT





THANK YOU FOR YOUR ATTENTION!