Industrial-Strength Formal Verification of RISC-

V Processors

Dr. Ashish Darbari Founder & CEO Axiomise



Verification Trends

Wilson research reports 2024

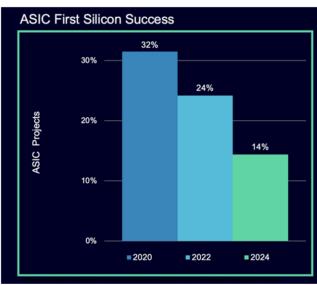


Fig. 1: Number of designs that are functionally correct and manufacturable is declining. Source: Siemens EDA/Wilson Research Group 2024 Functional Verification Study/DVCon

SYSTEMS & DESIGN

First-Time Silicon Success Plummets



Number of designs that are late increases. Rapidly rising complexity is the leading cause, but tools, training, and workflows need to improve.

MARCH 27TH, 2025 - BY: ED SPERLING

....

First-time silicon success is falling sharply due to rising complexity, the need for more iterations as chipmakers shift from monolithic chips to multi-die assemblies, and an increasing amount of customization that makes design and verification more time-consuming.

TECHNICAL PAPERS

Scalable And Energy Efficient Solution For Hardware-Based ANNs (KAUST, NUS)

MARCH 30, 2025 BY TECHNICAL
PAPER LINK

GPU Analysis Identifying Performance Bottlenecks That Cause Throughput Plateaus In Large-Batch Inference

MARCH 30, 2025 BY TECHNICAL PAPER LINK Strategies For Reducing The

Effective GaN/Diamond TBR



NEWSLETTER SIGNUP

SPONSORS



Axiomise Solutions

Making formal normal by building a tool agnostic layer of solutions

Training

1-2-3-4 days

Instructor-led

On-demand

Primer

Comprehensive

Methodology

Tool independent

Any duration Training follow up Methodology Strategy Planning Review Mentoring

Consulting

Services

Any duration Verification Strategy Verification Planning Execution Sign-off Weekly updates

Agile workflow

Apps

formalISA[®], footprint[™] Tool vendor independent Push button, easy set-up Find arch & uArch bugs Functional verification Safety, security, PPA Bug presence & absence

Consulting & Services

Formal verification at scale – turnkey services delivered on some of the projects

We have been carrying out functional verification of designs with over 1 billion gates with formal.

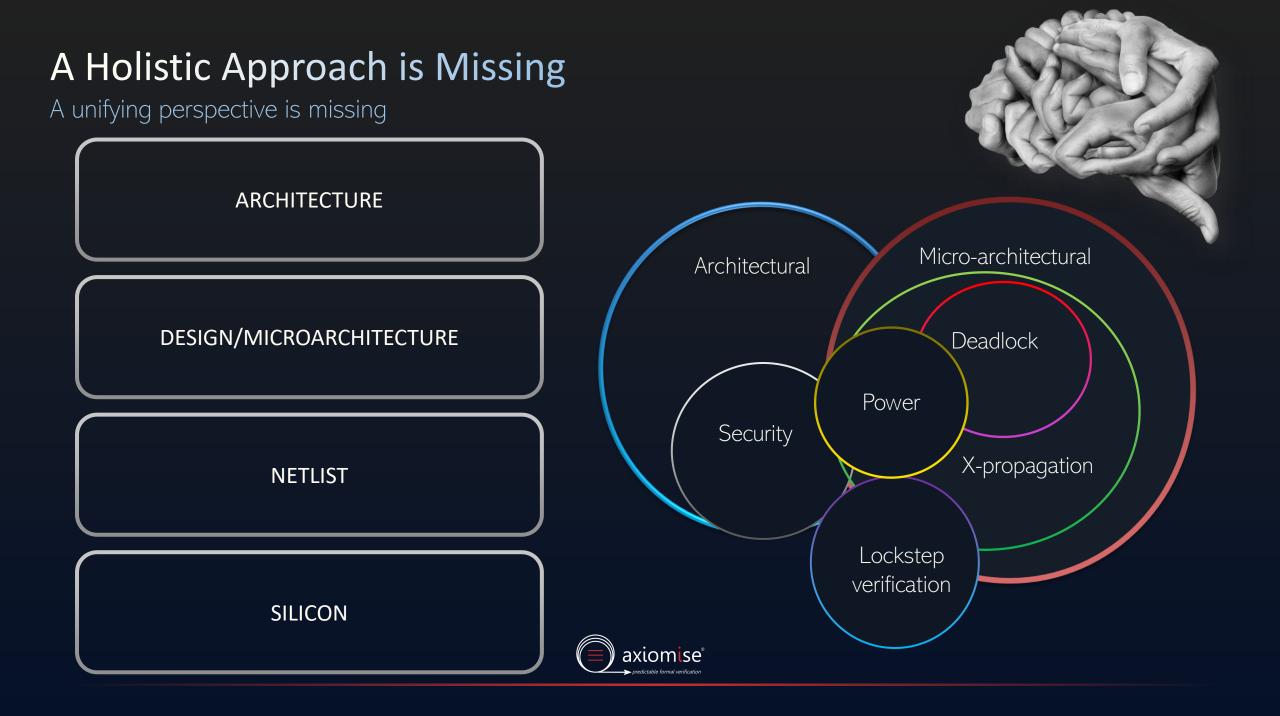
Our abstraction-powered methodologies work can find bugs in new and existing designs.

We also help with customers' post-silicon issues on designs previously verified by others.

AI/ML accelerator NoC Ethernet Switch Mixed-signal, low-power chip Power controller DMA controller Multi-threaded processor Bus bridges (AXI/CHI/OCP/TileLink) Cache sub-systems GPU blocks I2C/USB/HDMI/I2S

Why is processor verification hard? Why bugs escape to silicon?





Modern-day Processors

Massively optimised

Pipelining	Interlocking	Forwarding
Branches	Jumps	Exceptions
Stalls	Interrupts	Debug
Extensions	Clock gating	Arithmetic
Power	Safety	Security



Complex Control and Data Dependencies

And the cores have in-order or out-of-order behaviour?

Branches:

- Speculative branches
- Forward jumps, Backward jumps, Page size jumps, Page boundary jumps, Jumps across pages (same or different pages)

Back-to-back memory operations:

- Cache hit & cache misses
- Write-through stores
- Cache bypasses, atomics and cache coherency







Accelerating debug and sign-off for custom designs using exhaustive formal

Our Formal RISC-V Solution

Enables adoption of formal methods more widely

- 1. No test case to write
- 2. No manual checker to write
- 3. No verification code to be written
- 4. Exhaustively prove that all ISA instructions work as expected under all conditions What goes in our APP?
 - 1. Your RISC-V core
 - 2. Set up file
 - 3. Coverage specification

What comes out?

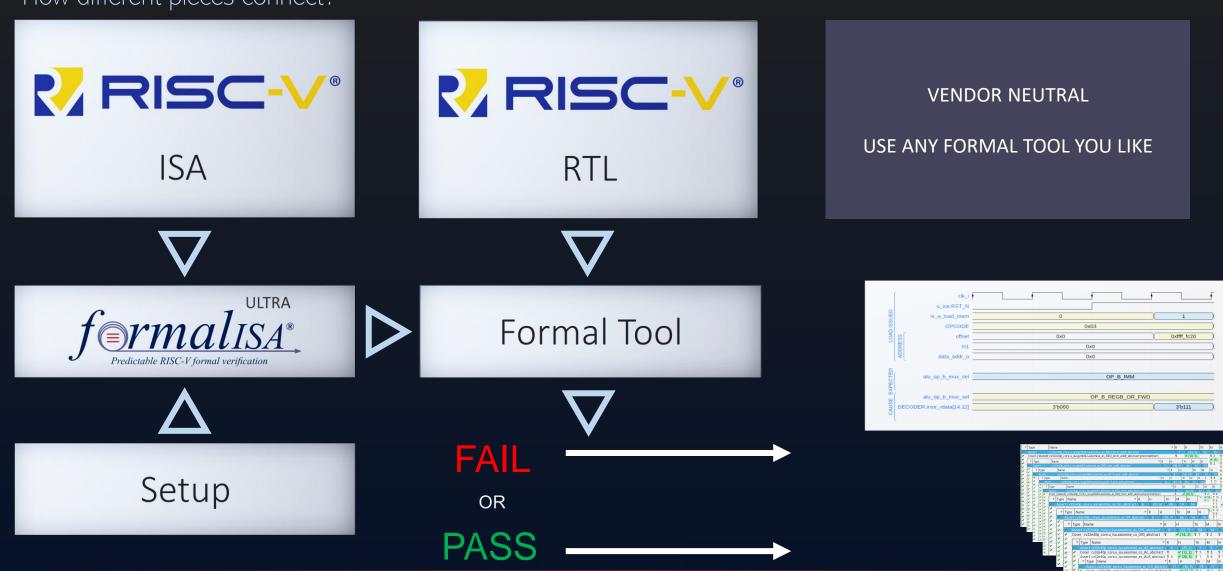
Exhaustive proofs that "mathematically" prove under all conditions:

- Each instruction in the ISA works always as expected
- Scenarios specified in the coverage specification can "always" happen
- Visualize that scenarios in the coverage specification "can happen"



formalISA How different pieces connect?





ibex

Complete democracy – use any tool you like

					svaju isajaxiomise ISA ADDI	0-15-17/1	noid
 Image: A set of the set of the	ibex_core.u_isa.axiomiseISA_JALR	• u_isa.axiomiseISA_SRLI	Assert 🖌	ibex_core.u_isa.inv_block_jal[4].axiomise_inv_isa_jal	- sva/u_isa/axioniise_ISA_ADDi	pass (4) pass (4)	hold
~	ibex_core.u_isa.axiomise_ISA_LUI	🗆 💿 💻 u_isa.axiomiseISA_ADDI	Assert 🖌	ibex_core.u_isa.inv_block_jalr[0].axiomise_inv_isa_jalr	sva/u_isa/axiomise_ISA_ANDI	pass (4)	hold
•		• u_isa.axiomiseISA_SRL	Assert 🗸	ibex_core.u_isa.inv_block_auipc[0].axiomise_inv_isa_auipc	sva/u_isa/axiomise_ISA_BEQ	pass (4)	hold
~	ibex_core.u_isa.axiomiseISA_OR	• u_isa.axiomiseISA_JALR	Assert 🗸	ibex_core.u_isa.inv_block_auipc[4].axiomise_inv_isa_auipc	sva/u_isa/axiomise_ISA_BGES	pass (4)	hold
-	ibex core.u isa.axiomise ISA ORI	u_isa.axiomiseISA_XORI	Assert 🗸	ibex_core.u_isa.inv_block_auipc[8].axiomise_inv_isa_auipc	sva/u isa/axiomise ISA BGEU	pass (4)	hold
~	Ibex_core.u_isa.axiomiseisa_oki	• u_isa.axiomiseISA_BEQ	Assert A	ibex_core.u_isa.inv_block_auipc[12].axiomise_inv_isa_auipc	sva/u isa/axiomise ISA BLTS	pass (5)	hold
~	ibex_core.u_isa.axiomise_ISA_SLL	u_isa.axiomiseISA_BNE	Assert A	ibex_core.u_isa.inv_block_auipc[20].axiomise_inv_isa_auipc	sva/u isa/axiomise ISA BLTU	pass (5)	hold
•		u_isa.axiomiseISA_SLTSI_SET_TO_1	Assert	ibex_core.u_isa.inv_block_auipc[24].axiomise_inv_isa_auipc	sva/u isa/axiomise ISA BNE	pass (5)	hold
 Image: A set of the set of the	ibex_core.u_isa.axiomiseISA_SLLI	• u_isa.axiomiseISA_BGEU	Assert	ibex_core.u_isa.inv_block_auipc[28].axiomise_inv_isa_auipc	sva/u isa/axiomise ISA JAL	pass (4)	hold
~	ibex core.u isa.axiomise ISA SLTSI SET TO 0	• u_isa.axiomiseISA_SLTSI_SET_TO_0	Assert	ibex_core.u_isa.axiomise_ISA_JAL	sva/u isa/axiomise ISA JAL ret address	pass (5)	hold
•		u_isa.axiomiseISA_XOR	Assert	ibex_core.u_isa.axiomise_ISA_JAL_ret_address	sva/u isa/axiomise ISA JALR	pass (4)	hold
✓	ibex_core.u_isa.axiomiseISA_SLTSI_SET_TO_1	• u_isa.axiomiseISA_SLTUI_SET_TO_1	Assert	ibex_core.u_isa.axiomise_ISA_JALR	sva/u_isa/axiomise_ISA_LUI	pass (4)	hold
	ibex_core.u_isa.axiomiseISA_SLTS_SET_TO_0	— 🗆 💿 💻 u_isa.axiomiseISA_OR	Assert	ibex_core.u_isa.axiomise_ISA_BEQ	sva/u_isa/axiomise_ISA_OR	pass (4)	hold
✓	IDex_core.d_isa.axiomiseibA_beis_bei_ro_o	• u_isa.axiomiseISA_ORI	Assert	ibex_core.u_isa.axiomise_ISA_BNE	sva/u_isa/axiomiseISA_ORI	pass (4)	hold
~	ibex_core.u_isa.axiomiseISA_SLTS_SET_TO_1	• u_isa.axiomiseISA_SLTUI_SET_TO_0	Assert	ibex_core.u_isa.axiomise_ISA_BGEU	sva/u_isa/axiomiseISA_SLL	pass (4)	hold
	iber and with evidening JCA CITUL CET TO O	💿 🔍 u_isa.axiomiseISA_ANDI	Assert	ibex_core.u_isa.axiomise_ISA_BLTU	sva/u_isa/axiomiseISA_SLLI	pass (4)	hold
✓	ibex_core.u_isa.axiomiseISA_SLTUI_SET_TO_0	• u_isa.axiomise_ISA_SLLI	Assert	ibex_core.u_isa.axiomise_ISA_LUI	sva/u_isa/axiomiseISA_SLTS_SET_TO_0	pass (4)	hold
~	ibex_core.u_isa.axiomiseISA_SLTUI_SET_TO_1	□ ◎ 💵 u_isa.axiomiseISA_SLTS_SET_TO_1	Assert	ibex_core.u_isa.axiomise_ISA_ADDI	sva/u_isa/axiomiseISA_SLTS_SET_TO_1	f (ass (5)	hold
•		_ ○ 🔍 u_isa.axiomise_ISA_SLTU_SET_TO_1	Assert	ibex_core.u_isa.axiomise_ISA_XORI	sva/u_isa/axiomiseISA_SLTSI_SET_TO_0	pass (4)	hold
 Image: A set of the set of the	ibex_core.u_isa.axiomiseISA_SLTU_SET_TO_0	• u_isa.axiomise_ISA_BLTS	Assert	ibex_core.u_isa.axiomise_ISA_ORI	sva/u_isa/axiomiseISA_SLTSI_SET_TO_1	pass (4)	hold
	ibex core.u isa.axiomise ISA SLTU SET TO 1	□ ◎ 💴 u_isa.axiomiseISA_SRAI	Assert	ibex_core.u_isa.axiomise_ISA_ANDI	sva/u_isa/axiomiseISA_SLTU_SET_TO_0	pass (4)	hold
~		□ ◎ = u_isa.axiomiseISA_SLTS_SET_TO_0	Assert	ibex_core.u_isa.axiomise_ISA_BLTS	sva/u_isa/axiomiseISA_SLTU_SET_TO_1	pass (5)	hold
~	ibex_core.u_isa.axiomiseISA_SRA	• u_isa.axiomise_ISA_SLTU_SET_TO_0	Assert	ibex_core.u_isa.axiomise_ISA_BGES	sva/u_isa/axiomiseISA_SLTUI_SET_TO_0	pass (4)	hold
-	ibex core.u isa.axiomise ISA SRAI	□ ◎ ■ u_isa.axiomise_ISA_JAL	Assert	ibex_core.u_isa.axiomise_ISA_SLTSI_SET_TO_1	sva/u_isa/axiomiseISA_SLTUI_SET_TO_1	pass (4)	hold
 	IDEX_COTE.d_ISB.BXIOTIISE_ISA_SKAI	□ ◎ ■ u_isa.axiomise_ISA_LUI	Assert	ibex_core.u_isa.axiomise_ISA_SLTSI_SET_TO_0	sva/u_isa/axiomiseISA_SRA	pass (4)	hold
~	ibex core.u isa.axiomise ISA SRL	• u_isa.axiomise_ISA_BGES	Assert	ibex_core.u_isa.axiomise_ISA_SLTUI_SET_TO_1	sva/u_isa/axiomiseISA_SRA1	pass (4)	hold
•		O u_isa.axiomise_ISA_SRA SRA S	Assert	ibex_core.u_isa.axiomise_ISA_SLTUI_SET_TO_0	sva/u_isa/axiomiseISA_SRL	pass (4)	hold
 Image: A set of the set of the	ibex_core.u_isa.axiomiseISA_SRLI	• v_isa.axiomise_ISA_BLTU	Assert	ibex_core.u_isa.axiomise_ISA_SLLI	sva/u_isa/axiomise_ISA_SRLI	pass (4)	hold
~	ibex_core.u_isa.axiomise_ISA_SUB	— □ ◎ 💴 u_isa.axiomiseISA_ADD	Assert	ibex_core.u_isa.axiomise_ISA_SRLI	sva/u_isa/axiomiseISA_SUB	pass (4)	hold hold
•		□ ◎ 💵 u_isa.axiomiseISA_AND	Assert	ibex_core.u_isa.axiomise_ISA_SRAI	sva/u_isa/axiomiseISA_XOR	pass (4)	hold
 Image: A set of the set of the	ibex_core.u_isa.axiomiseISA_XOR	□ ◎ ■ u_isa.axiomise_ISA_SUB	Assert	ibex_core.u_isa.axiomise_ISA_SLL	sva/u_isa/axiomiseISA_XORI	pass (4) pass (4)	hold
	ibex core.u isa.axiomise ISA XORI	□ ◎ ■ u_isa.axiomise_ISA_SLL	Assert	ibex_core.u_isa.axiomise_ISA_SRL	sva/u_isa/axiomiseHISA_AUIPC	pass (4) pass (4)	hold
 Image: A set of the set of the	IDEX_COLE.U_ISB.BXIOHIISE_ISA_XORI				sva/u_isa/axiomiseHISA_JALR2	pass (4)	noid

axiomise









Formal verification Bugs and Proofs





	ibex	zeroriscy	cv32e40p		WARP-V		Cheriot-ibex
Pipeline stages	2-stage	2-stage	4-stage	6-stage	4-stage	2-stage	2
No. of issues	65	77	5	30	30	30	6
Previously verified	Yes	Yes	No	Yes	Yes	Yes	Yes
How was it previously verified?	Simulation	Simulation	Simulation & Formal	Formal	Formal	Formal	Simulation & Formal
Time taken to find issues	< 30 seconds	< 30 seconds	< 30 seconds	< 30 seconds	< 30 seconds	< 30 seconds	<1 min
Nature of analysis and issues	Microarchitectural Deadlocks and Architectural	Microarchitectural Deadlocks and Architectural	Architectural	Architectural	Architectural	Architectural	Corner-case bugs
When was the issue found?	2019	2019	2020	2021	2021	2021	2024



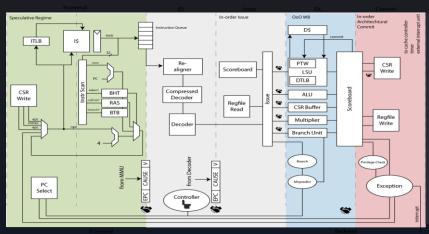
cv32e40p

32-bit, 4-stage in-order pipeline

Ţ	Тур	e	N	Name				<u>.</u>	₹ R	Н	Tr	ri M	N										
~	Asse	ert			ore.u_isa.genblk2.axior				?	7 💡 (44, 6) 💡	[°] 52 🖋	15 💡 18	8									
~	Cov	er (re	lated) c	v32e40p_co	ore.u_isa.genblk2.axior	mise_as_BEQ_inst	.r_addr_abstract:pr	econdition1	?	🖌 (1	10, 1)		2 ? 8										
	Ψ.	Туре		Name				\	н	Tri	M N		15 ? 18 2 ? 8										
		Assert	t	cv32e40p_co	ore.u_isa.genblk3.axiomise	e_as_BNE_instr_add	lr_abstract	2	7 💡 (40,	, 6) 🖋 54	2 12 2	14	2 78										
v		ΨT	Туре	Name					Ÿ R	Н	Tri N	M N	26										
~	-		Assert		10p_core.u_isa.genblk6.ax				28	2 (58, 10) 🗸 80 📍	2 12 💡	14 ? 19	9									
~	•				10p_core.u_isa.genblk6.ax			condition1	?	✓ (12, 2)		? 2 ?	8 ? 8										
~	× 1		Cover Assert		10p_core.u_isa.genblk6.ax 10p_core.u_isa.genblk6.ax				? ? 9	 ✓ (14, 2) ² (66, 24) 	¥ 1 1) ✓ 76 1		10 ? 7										
*	×				10p_core.u_isa.genblk6.ax			ition1	?	 ✓ (12, 2) 	, , , , , , , ,		¥ 10	_									
	 I 			7 Туре	Name				ΨR	н	Tri M	N	5 28										
	~		Assert 🧹	Assert	cv32e40p_core.u_isa.ger ed) cv32e40p_core.u_isa.ger			nuccondition 1	? 8	? (58, 10)	✓ 80 💡 1		16 8 ? 20	0									
~			Cover 🖌	Cover (related	cv32e40p_core.u_isa.ger			recondition	?		? 1 ? 2 ? 1 ? 2		0										
~	-	🖌 A	Assert 🛩	Assert	cv32e40p_core.u_isa.ger				? 9	2 (66, 24)	🖌 76 🛭 🕆 8	🔋 16	18 🖌 20	6									
~	•		Cover 🗸	Cover (related	ed) cv32e40p_core.u_isa.ger cv32e40p_core.u_isa.ger			ndition1	?	 ✓ (12, 2) ✓ (14, 2) 	? 1	2 5	12										
*			Cover Assert	Assert	cv32e40p_core.u_isa.axi	iomise_as_BLTS_abstr	ract		28	2 (66, 11)	🗸 80 💡 1-	4 💡 16	20	_									
~	-		Cover	Cover (related	ed) cv32e40p_core.u_isa.axi cv32e40p_core.u_isa.axi					 ✓ (12, 2) ✓ (16, 3) 			20										
v	 Image: A set of the set of the		Assert 🖌	Ass Tvp	······	011130_00_8218_0880				▼ R		Tri	М	N									
			Cover 🖌	Cov Ass		ore.u_isa.genblk2.axi	omise_as_BEQ_instr_ac	ddr_abstract		?	7 ? (44	4, 6) 💡	52 🖋 15	? 18									
	È		Cover 🖌	Ass 🖌 Cov	ver (related) cv32e40p_co	ore.u_isa.genblk2.axid	omise_as_BEQ_instr_ad	ddr_abstract:preco	ondition1	?				? 8 ? 19									
			Cover 🖌	Cov 🗸 Ass Ass 🗸 Cov	ver (related) cv32e40p co		omice as ISA REO inc	tr addr				1	/// // 15										
		🖌 C	Cover 🗸	Cov 🗸 Cov	· · · · · · · · · · · · · · · · · · ·	🛚 Type 🛛 Na 🖙	^r Type Name				Ŷ	R	H	Tri	M	N							
			*	Cov 🗸 Cov		Assert cv: 🖌	Assert cv32e40		-is-using				le (20			4.40	1	1					
			✓	Cov Ass Cov Cov	sert cv32e4 ver (related) cv32e4	Cover cv: 🗸	Cover cv32e	♥ Type Na	me					Y	R I	Н	Tri	M	N				
			~	Cov 🖌 Cov	ver cv32e4	Assert cv: 🗸	Assert cv32	Assert cv3	32e40p (ore.u is	a.axiom	ise as	ORI ab	stract	? 9 4	2 (22, 7)	? 58	2 14	✓ 17				
				✓ Ass		Cover cv: 🖌	Cover cv32	Cover cv3								(14, 2)		22	? 12				
				✓ Cov✓ Cov	ver (related) cv32e4	Assert cv:	Assert cv32	Assert cv3			and do in	100_00		beruce		(14, 2)	• •						
				🖌 Ass	sert cv32e4	Cover cv:	Cover cv32	Cover cv3			🛛 Тур	e Nai	me				-	₹ R	Н	Tri	M	N	
				✓ Cov✓ Ass	ver (related) cv32e4	Cover cv:	Assert cv32	Assert cv3					2e40p c	ore.u isa	axiomis	e as IAL	abstract	24	2 (20, 2)	21	23	✓ 22	
					ver (related) cv32e4	Assert cv:	Cover cv32	Cover cv3						ore.u_isa					✓ (12, 2)	21	? 3	? 10	
				🖌 Cov	ver cv32e4	Cover cv:	Assert cv32	Assert cv3						ore.u_isa					✓ (20, 3)	21	26	2 14	
				✓ Cov		Assert cv:	Cover cv32							ore u isa					✓ (20, 3)✓ (12, 2)		23	2 10	
				✓ Cov✓ Cov	vei cv52e4	Cover cv:	Assert cv32	Cover cv3				ert c		Name				-	ΨR	Н	Tri	M	N
							Cover cv32	Assert cv3			 Cov 				On core	u isa avio	mise as	LUI abs	tract 💡 11	× (36		15 2 11	7 7 17
								Cover cv3				ert c 🗸	hanned	r cv32e4		_		_		✓ (30,✓ (14,		22	? 12
								Assert cv3				/er c ✓		rt cv32e4					? 10	? (36,	-/ -	50 2 14	
								Cover cv3	2e40p_c	:ore.u_i	COV			r cv32e4					2	 ¹ (30, ¹ (14,	- / -	1 ? 2	
												Ě	2070		op_core.	a_150.07101		10/1_101		· (14,	-/ •	• 2	• 12

CVA6

64-bit six-stage, in-order issue, out-of-order execution, in-order commit



From the OPENHW Group Page

CVA6 is a 6-stage, single issue, in-order CPU which implements the 64-bit RISC-V instruction set. It fully implements I, M, A and C extensions as specified in Volume I: User-Level ISA V 2.3 as well as the draft privilege extension 1.10. It implements three privilege levels M, S, U to fully support a Unix-like operating system. Furthermore, it is compliant to the draft external debug spec 0.13. It has configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer and branch history table). The primary design goal was on reducing critical path length.

Image: Section of the section of th														
Property Table Curdon Buttorn Search Image: State Property Table Image: State Property Table Image: State Curdon Usable, abstraction State State Property Table Image: State Curdon Usable, abstraction State State Property Table Image: State Curdon Usable, abstraction State State Property Table Image: Assent Curdon Usable, abstraction Basts RTYPE, as ISA, AND, bits, abstract N (60) Infinite Zatable, 0 sembedded> Assent Curdon Usable, abstraction Basts RTYPE, as ISA, SUB, bits, abstract Tri (66) Infinite Zatable, 0 sembedded> Assent Curdon Usable, abstraction Basts, SUB, bits, abstract Tri (66) Infinite Zatable, 0 sembedded> Assent Curdon Usable, RTYPE, as ISA, SUB, bits, abstract Tri (60) Infinite Zatable, 0 sembedded> Assent Curdon Usable, RTYPE, as ISA, SUB, bits, abstract Tri (66) Infinite Zatable, 0 sembedded> Assent Curdon Usable, RTYPE, as ISA, SUB, bits, abstract Tri (66) Infinite Zatable, 0 sembedded> Assent Curdon Usable, RTYPE, as ISA, SUB, bits, abstract Tri (66)														
Image: I														
Property Table Yeight of the second secon														
Yes No filter T-Filter on name Yes No filter Vestion Regine Vestion Bound Time Task Task Yes Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_AND_bits abstract N (60) Infinite Tiskel Yes Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_ADD_bits abstract Tri (66) Infinite Tiskel cembedded> Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_ADD_bits abstract Tri (88) Infinite Tiskel cembedded> Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_XDB_bits abstract Tri (64) Infinite Tiskelded> Assent Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_XDB_bits abstract Tri (60) Infinite Tiskelded> Assent Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_XDB_bits abstract Tri (60) Infinite Tiskelded> Assent Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_XDB_bits abstract Tri (66) Infinite Tiskelded> Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_XDB_bits abstract Tri (66) Infinite Tiskelded> Assent Cva6u (sa bit abstract(0) BASE_RTYPE as [SA_XDB_bits abstract Tri (66														
V Type V Rame V Type V Rame V Type V Rame	Property Table													
Assert cva6.u (sa.bl.) bstract(0) EASE; RTYPE as (SA.AND bits abstract N (co) Infinite 234494.0 combedded> Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.AND bits abstract Tr (66) Infinite 234494.0 combedded> Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.OD bits abstract Tr (68) Infinite 23492.5 cembedded> 4 Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.SUB) bits abstract Tr (68) Infinite 21492.5 cembedded> 4 Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.SUB) bits abstract Tr (60) Infinite 12490.5 cembedded> 4 Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.SUB) bits abstract Tr (60) Infinite 12499.4 cembedded> 4 Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.SUL) bits abstract Tr (74) Infinite 121499.4 cembedded> 4 Assert cva6.u (sa.bl.) abstract(0) EASE; RTYPE as (SA.SUL) bits abstract Tr (165) Infinite 121499.4 cembedded> 4 Asseset cva6.u (sa.bl.)	Ng ≈ Ng No filter y γ-Filter on name													
Besent cva6u (sa bit abstractio) BASE_RTYPE as ISA AND bits abstract It (co) Infinite 234484.0 scembadded> 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, AND bits abstract Tri (66) Infinite 234484.0 scembadded> 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, OD bits abstract Tri (66) Infinite 116840.6 cembadded> 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, SUB bits abstract Tri (68) Infinite 12890.5 cembadded> & 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, SUB bits abstract Tri (68) Infinite 12890.5 cembadded> & 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, SUB bits abstract Tri (60) Infinite 12890.5 cembadded> & 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, SUD bits abstract Tri (60) Infinite 121499.4 cembedded> & 4 Assent cva6u (sa bit abstractio) BASE_RTYPE as ISA, SUB bits abstract Tri (65) Infinite 121499.4 cembedded> & <th colspan="14">g <u>/</u> ▼ Type ▼ Name</th>	g <u>/</u> ▼ Type ▼ Name													
Assert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.JOD.bis_abstract Tri (88) Infinite 224192.5 cembedded> VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SUB.bis_abstract Tri (80) Infinite 224092.5 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SUB.bis_abstract Tri (80) Infinite 12409.0 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SUS.bis_abstract Tri (80) Infinite 12409.4 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SUS.bis_abstract Tri (66) Infinite 124194.4 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SUL bis_abstract Tri (66) Infinite 122127.4 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SR.bis_abstract Tri (62) Infinite 1203534.1 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SR.bis_abstract Tri (62) Infinite 1203534.1 cembedded> & VASsert cva6.u jsa.bi.abstract(0).BASE_RTYPE as jSA.SR.bis_abstract Tri (63) Infinite 109057.3 c	&	ded>	&	Analysis Session										
Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_DD_bits_abstract Tri (68) Infinite 224192.5 cembedded> Massert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_DD_bits_abstract Tri (64) Infinite 22409.50 cembedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SUB_bits_abstract Tri (64) Infinite 12409.50 cembedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SUTS_bits_abstract Tri (64) Infinite 124194.4 cvmbedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SUTS_bits_abstract Tri (66) Infinite 124194.4 cembedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SUL bits_abstract Tri (65) Infinite 122197.4 cembedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SNL bits_abstract Tri (62) Infinite 120327.3 cembedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SNL bits_abstract Tri (63) Infinite 120327.4 cembedded> & Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SNL bits_abstract Tri (63) Infinite 11997.0 <	&	ded>	&	Analysis Session										
Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA JOR_bits_abstract Tri (80) Infinite 145447.5 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_SUS bits_abstract Tri (74) Infinite 186137.6 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_SUL bits_abstract Tri (66) Infinite 121499.4 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_SUL bits_abstract Tri (66) Infinite 121297.4 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_SRL bits_abstract Tri (66) Infinite 1203541.1 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_SRL bits_abstract Tri (62) Infinite 120357.3 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_SRL bits_abstract Tri (63) Infinite 11997.0 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_ADDI bits_abstract Tri (63) Infinite 11997.0 cembedded> Assert Assert cva6.u jsa.bit.abstract101 BASE_RTYPE as JSA_ADDI bits_abstract Tr	&	ded>	&	Analysis Session										
Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_SL_bits_abstract Tri (65) Infinite 127127.4 <=mbedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_SL_bits_abstract Tri (70) Infinite 20323.11 <=mbedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_SNLbits_abstract Tri (62) Infinite 90267.3 <embedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_AND_bits_abstract Tri (62) Infinite 90267.3 <embedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_AND_bits_abstract Tri (65) Infinite 10975.0 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_ADD_bits_abstract Tri (66) Infinite 10987.0 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_ADD_bits_abstract Tri (66) Infinite 10985.1.7 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_ADD_bits_abstract Tri (66) Infinite 10985.1.7 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_SUB_bits_abstract Tri (67) Infinite 10985.1.7 <embedded> Assert<th>&</th><th>ded> &</th><th>&</th><th>Analysis Session</th></embedded></embedded></embedded></embedded></embedded></embedded></embedded>	&	ded> &	&	Analysis Session										
Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SL_bits_abstract Tri (65) Infinite 127127.4 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SL_bits_abstract Tri (70) Infinite 20237.3 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SA_bits_abstract Tri (62) Infinite 90267.3 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_AND_bits_abstract Tri (62) Infinite 90267.3 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_AND_bits_abstract Tri (63) Infinite 19982.0 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_ADDL_bits_abstract Tri (66) Infinite 10982.0 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_ADDL_bits_abstract Tri (66) Infinite 10982.0 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_ADDL_bits_abstract Tri (66) Infinite 10982.1 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_ADDL_bits_abstract Tri (66) Infinite 10992.4 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_STLUB is_bastract Tri (66) Infinite 2097.73 =mbedded>	&	ded>	&	Analysis Session										
Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SL_bits_abstract Tri (65) Infinite 127127.4 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SL_bits_abstract Tri (70) Infinite 2023.31.4 <=mbedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_SA_bits_abstract Tri (62) Infinite 90267.3 <embedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_AND_bits_abstract Tri (62) Infinite 90267.3 <embedded> Assert cva6u_isa.bit_abstract(0).BASE_RTYFE_as_ISA_AND_bits_abstract Tri (63) Infinite 10982.0 <embedded> Assert cva6u_isa.bit_abstract(0).BASE_ITYFE_as_ISA_ADD_bits_abstract Tri (66) Infinite 10982.0 <embedded> Assert cva6u_isa.bit_abstract(0).BASE_ITYFE_as_ISA_ADD_bits_abstract Tri (66) Infinite 10982.1 <embedded> Assert cva6u_isa.bit_abstract(0).BASE_ITYFE_as_ISA_ADD_bits_abstract Tri (66) Infinite 10982.1 <embedded> Assert cva6u_isa.bit_abstract(0).BASE_ITYFE_as_ISA_STUB_bits_abstract Tri (66) Infinite 10982.1 <embedded> Assert</embedded></embedded></embedded></embedded></embedded></embedded></embedded>				Analysis Session										
Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_SL_bits_abstract Tri (65) Infinite 127127.4 <=mbedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_SL_bits_abstract Tri (70) Infinite 20323.11 <=mbedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_SNLbits_abstract Tri (62) Infinite 90267.3 <embedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_AND_bits_abstract Tri (62) Infinite 90267.3 <embedded> Assert cva6.u jsa.bit_abstract(0) ASS_ETYPE_as_JSA_AND_bits_abstract Tri (65) Infinite 10975.0 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_ADD_bits_abstract Tri (66) Infinite 10987.0 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_ADD_bits_abstract Tri (66) Infinite 10985.1.7 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_ADD_bits_abstract Tri (66) Infinite 10985.1.7 <embedded> Assert cva6.u jsa.bit_abstract(0) BASE_TYPE_as_JSA_SUB_bits_abstract Tri (67) Infinite 10985.1.7 <embedded> Assert<th>&</th><th>ded></th><th>&</th><th>Analysis Session</th></embedded></embedded></embedded></embedded></embedded></embedded></embedded>	&	ded>	&	Analysis Session										
Assert cva6.u_isa.bit_abstract(0).BASE_RTYPE_as_ISA_SRA_bits_abstract Tri (62) Infinite 90267.3 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_RTYPE_as_ISA_AND) bits_abstract Tri (63) Infinite 119975.0 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_AND) bits_abstract Tri (65) Infinite 10982.4 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_ANDI_bits_abstract Tri (66) Infinite 10982.4 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_NOL bits_abstract Tri (66) Infinite 10982.4 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_NOL bits_abstract Tri (66) Infinite 10987.7 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_SUL bits_abstract Tri (67) Infinite 12977.4 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_SULU bits_abstract Tri (66) Infinite 20676.8 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_ISA_SULU bits_abstract Tri (65) Infinite 20676.8 cembedded> Assent cva6.u_isa.bit_abstract(0).BASE_ITYPE_as_IS				Analysis Session										
Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_ANDL bits_abstract Tri (63) Infinite 11997.0 combedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_ANDL bits_abstract Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_NORL bits_abstract Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_NORL bits_abstract Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (66) Infinite 206768.8 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 206768.8 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 34304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 4304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 34304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64304.9 <embedded> Assert cva6.u jsa.bit_abstract(0).BASE_ITYPE_as_ISA_SIUL bits_abstract N (56) Infinite 64</embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded>				Analysis Session										
Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ORI_bits_abstract Tri (65) Infinite 105982.4 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ADDL bits_abstract Tri (66) Infinite 109582.4 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ADDL bits_abstract Tri (66) Infinite 109582.4 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ADDL bits_abstract Tri (66) Infinite 119922.4 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SUSI_bits_abstract Tri (67) Infinite 119927.4 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SUSI_bits_abstract N (56) Infinite 121977.4 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SUU_bits_abstract N (56) Infinite 20676.8 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SUU_bits_abstract Tri (65) Infinite 84304.9 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SUL_bits_abstract Tri (66) Infinite 65054.7 <embedded> Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_I</embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded>				Analysis Session										
Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_ADD_bits_abstract Tri (66) Infinite 109651.7 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_XORI_bits_abstract Tri (66) Infinite 11992.4 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_SURI_bits_abstract Tri (67) Infinite 11997.74 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_SULU_bits_abstract Tri (67) Infinite 2077.74 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_SULU_bits_abstract N (56) Infinite 206768.8 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_SULU_bits_abstract Tri (65) Infinite 84304.9 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_SULU_bits_abstract Tri (66) Infinite 6054.7 <embedded> & Assert cva6.u_isa.bit_abstract(01.BASE_ITYPE_as_ISA_SULU_bits_abstract Tri (66) Infinite 6054.7 <embedded> &</embedded></embedded></embedded></embedded></embedded></embedded></embedded></embedded>				Analysis Session										
Assert cva6.u jsa.bit_abstract(0) BASE_ITYPE as_ISA_XORI_bits_abstract Tri (64) Infinite 118922.4 < embedded> Assert cva6.u jsa.bit_abstract(0) BASE_ITYPE as_ISA_SISL bits_abstract Tri (67) Infinite 118922.4 < embedded> Assert cva6.u jsa.bit_abstract(0) BASE_ITYPE as_ISA_SIL bits_abstract Tri (67) Infinite 206768.8 embedded> Assert cva6.u jsa.bit_abstract(0) BASE_ITYPE as_ISA_SILU_bits_abstract N (56) Infinite 206768.8 embedded> Assert cva6.u jsa.bit_abstract(0) BASE_ITYPE as_ISA_SILU_bits_abstract Tri (65) Infinite 84304.9 embedded> Assert cva6.u jsa.bit_abstract(0) BASE_ITYPE as_ISA_SILU_bits_abstract Tri (66) Infinite 84304.9 embedded>				Analysis Session										
				Analysis Session										
Assent cva6.u_isa.bit_abstract(0].8ASE_ITYPE_as_ISA_SILU_bits_abstract N (56) Infinite 206768.8 <embedded> & Assent cva6.u_isa.bit_abstract(0].8ASE_ITYPE_as_ISA_SILU_bits_abstract Tri (65) Infinite 206768.8 <embedded> & Assent cva6.u_isa.bit_abstract(0].8ASE_ITYPE_as_ISA_SILU_bits_abstract Tri (65) Infinite 8303.4 <embedded> & Assent cva6.u_isa.bit_abstract(0].8ASE_ITYPE_as_ISA_SILU_bits_abstract Tri (66) Infinite 60504.7 cembedded> &</embedded></embedded></embedded>				Analysis Session										
Assert cva6.u_isa.bit_abstract(0].BASE_ITYPE_as_ISA_SLIL_bits_abstract Ya65) Infinite 84304.9 <embedded> Assert cva6.u_isa.bit_abstract(0].BASE_ITYPE_as_ISA_SRIL_bits_abstract Tri (66) Infinite 65054.7 <embedded></embedded></embedded>				Analysis Session										
🖌 Assert cva6.u_isa.bit_abstract[0].BASE_TYPE_as_ISA_SRL_bits_abstract Tri (66) Infinite 65054.7 <embedded></embedded>				Analysis Session										
				Analysis Session										
				Analysis Session										
🖌 Assert cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SRAI_bits_abstract Tri (61) Infinite 60750.6 < embedded>	&	ded> &	&	Analysis Session										
				Analysis Session										
🖌 Assert cva6.u_isa.bit_abstract(0).BASE_UTYPE_as_ISA_AUIPC_bits_abstract Tri (67) Infinite 93899.4 <embedded></embedded>	&	ded>	&	Analysis Session										



CVA6

64-bit six-stage, in-order issue, out-of-order execution, in-order commit

			D Issue D Commit								_
File	e <u>E</u> dit	View Design	Reports Application Window Help								
V	Form	nal Property V.	· •								
- File		Design Se	etup————————————————————————————————————								
6	2 🐼	🖻 🔻 🚅 🐾 🖡	🕾 🕅 🕈 🍰 ൽ 🖬 🖬 🕗 🖸 🔯 🧳 🕼 🔹 🖉								
Prop	erty Ta	ble									
		No filter	▼ ∏- Filter on name								
	••• 100				-		1			1	
es	$\Delta \mathbf{P}$	Type 🍸	Name 🏾 🕅	Engine	♥ Bound	Time	Task	8 8	Traces	Source	Y
Properties	I 1	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_AND_bits_abstract	N (60)	Infinite		<pre><embedded></embedded></pre>	&	(Analysis Sessio	n
pop		Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_OR_bits_abstract	Tri (66)	Infinite	116840.6	<pre>embedded></pre>	&	0	Analysis Session	n
<u> </u>	 ✓ 	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_ADD_bits_abstract	Tri (88)	Infinite	234192.5	<embedded></embedded>	&	0	Analysis Session	n
8	~	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_SUB_bits_abstract	Tri (64)	Infinite	128905.0	<embedded></embedded>	&	0	Analysis Session	n
Covergroups	~	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_XOR_bits_abstract	Tri (80)	Infinite	145447.5	<embedded></embedded>	&	0	Analysis Session	n
1 Dia	~	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_SLTS_bits_abstract	Tri (74)	Infinite	186137.6	<pre>embedded></pre>	&	0	Analysis Session	n
Š	 ✓ 	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_SLTU_bits_abstract	Tri (66)	Infinite	121499.4	<embedded></embedded>	&	0	Analysis Session	n
<u> </u>	 ✓ 	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_SLL_bits_abstract	Tri (65)	Infinite	127127.4	<embedded></embedded>	&	0	Analysis Session	n
	🖌 🕴	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_SRL_bits_abstract	Tri (70)	Infinite	203534.1	<embedded></embedded>	&	0	Analysis Session	n
	 ✓ 	Assert	cva6.u_isa.bit_abstract[0].BASE_RTYPE_as_ISA_SRA_bits_abstract	Tri (62)	Infinite	90267.3	<embedded></embedded>	&	0	Analysis Session	n
	~	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ANDI_bits_abstract	Tri (63)	Infinite	119875.0	<embedded></embedded>	&	0	Analysis Session	n
	~	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ORI_bits_abstract	Tri (65)	Infinite	105982.4	<embedded></embedded>	&	0	Analysis Session	n
	~	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_ADDI_bits_abstract	Tri (66)	Infinite	109651.7	<embedded></embedded>	&	0	Analysis Session	n
	🖌 🕴	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_XORI_bits_abstract	Tri (64)	Infinite	118922.4	<embedded></embedded>	&	0	Analysis Session	n
	🖌 🕴	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SLTSI_bits_abstract	Tri (67)	Infinite	171977.4	<embedded></embedded>	&	0	Analysis Session	n
	🖌 🕴	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SLTUI_bits_abstract	N (56)	Infinite	206768.8	<pre>embedded></pre>	&	0	Analysis Session	n
	🖌 🕴	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SLLI_bits_abstract	Tri (65)	Infinite	84304.9	<embedded></embedded>	&	0	Analysis Session	n
	🖌 🕴	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SRLI_bits_abstract	Tri (66)	Infinite	65054.7	<embedded></embedded>	&	0	Analysis Session	n
	~	Assert	cva6.u_isa.bit_abstract[0].BASE_ITYPE_as_ISA_SRAI_bits_abstract	Tri (61)	Infinite	60750.6	<pre>embedded></pre>	&	0	Analysis Session	n
	~	Assert	cva6.u_isa.bit_abstract[0].BASE_UTYPE_as_ISA_LUI_bits_abstract	N (55)	Infinite	120613.1	<embedded></embedded>	&	0	Analysis Session	n
	~	Assert	cva6.u_isa.bit_abstract[0].BASE_UTYPE_as_ISA_AUIPC_bits_abstract	Tri (67)	Infinite	93899.4	<embedded></embedded>	&	(Analysis Session	n



Reporting Scheduler and Reporter for Formal SURF

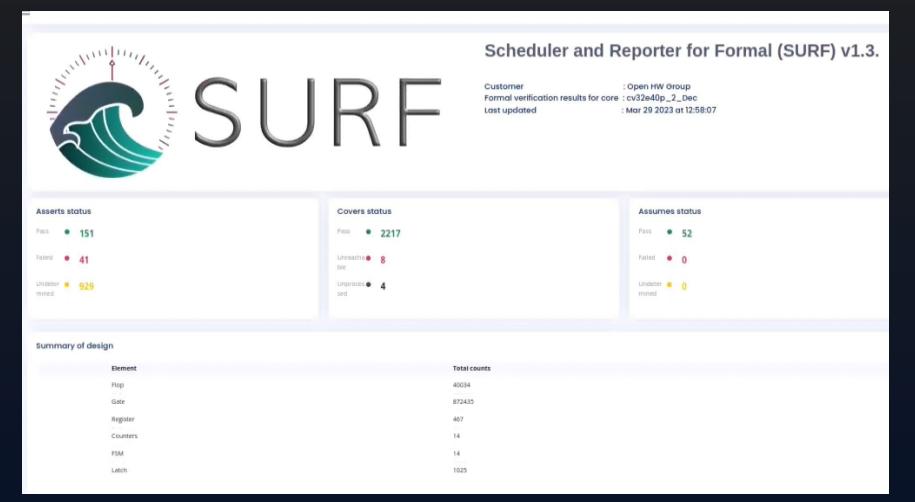


RISC-V

Sermative	=			
Overview exercer uses Assertices Covers Covers Assumptions	SU	Scheduler and R Customer romal verification results for core Last updated	eporter for Formal (SURF) v1.3. : 00pm HW 070up : 0x32940p, 2, 0xc war 29 2023 of 125807	
	Asserts status Pras • 151 Faire • 41 Unitariar • 929 mass	Covers status Pozz • 2217 Locatione 8 Locatione 4 Locatione 4	Assumes status Pres • 52 Faird • 0 Universe • 0 Universe • 0	Overview of total assumes, asserts, and covers
	Summary of design Exerce Hop Cate Register Countern TSM Later	Total courts 4034 87345 407 407 14 14 1625		Distributions of engines (Devergence of desertors
	Summary of testbench Remet Any Gate Cauter F3M Latin	Tetid counts 27347 783218 238 3 0		
	LAUT			Distributions of engines (Concentrate or concert



RISC-V





Example reporting bugs

formalise	=					
88 Overview						
REPORT LISTS	Covers					
C ASSERTIONS	No.	Instruction type	Property label	Cover status	Proof time	Engine
© COVERS	1,	BASE ITYPE	co_ISA_ADDI_abstract	COVERED	0.08	Hts
	2.	BASE ITYPE	co_ISA_ADDI_abstract_JG	COVERED	0.38	Hts
	3.	BASE ITYPE	co_ISA_ADDI	COVERED	0.07	Bm
	4.	BASE ITYPE	co_ISA_ADDI_JG	COVERED	0.68	Ht
	5.	BASE ITYPE	co_ISA_XORI_abstract	COVERED	0.07	Bm
	6.	BASE ITYPE	co_ISA_XORI_abstract_JG	COVERED	0.72	Ht
	7.	BASE ITYPE	co_ISA_XORI	COVERED	0.07	Bm
	8.	BASE ITYPE	co_ISA_XORI_JG	COVERED	0.72	Ht
	9.	BASE ITYPE	co_ISA_ORI_abstract	COVERED	0.07	Bm
	10.	BASE ITYPE	co_ISA_ORI_abstract_JG	COVERED	0.76	Ht
	11.	BASE ITYPE	co_ISA_ORI	COVERED	0.07	Bm
	12.	BASE ITYPE	co_ISA_ORI_JG	COVERED	0.76	Ht
	13.	BASE ITYPE	co_ISA_ANDI_abstract	COVERED	0.07	Bm
	14.	BASE ITYPE	co_ISA_ANDI_abstract_JG	COVERED	0.80	Ht
	15.	BASE ITYPE	co_ISA_ANDI	COVERED	0.07	Bm
	16.	BASE ITYPE	co_ISA_ANDI_JG	COVERED	0.80	Ht
	17.	BASE ITYPE	co_ISA_SLTI_SET_TO_1_abstract	COVERED	0.07	Bm
	18.	BASE ITYPE	co_ISA_SLTI_SET_TO_1_abstract_JG	COVERED	0.83	Ht
	19.	BASE ITYPE	co_ISA_SLTI_SET_TO_1	COVERED	0.07	Bm
	20.	BASE ITYPE	co_ISA_SLTI_SET_TO_1_JG	COVERED	0.83	Ht
	21.	BASE ITYPE	co_ISA_SLTI_SET_TO_0_abstract_JG	COVERED	0.87	Ht
	22.	BASE ITYPE	co_ISA_SLTI_SET_TO_0_JG	COVERED	0.87	Ht
	23.	BASE ITYPE	co_ISA_SLTI_SET_TO_0_abstract	COVERED	0.07	Bm
	24.	BASE ITYPE	co_ISA_SLTI_SET_TO_0	COVERED	0.07	Bm
	25.	BASE ITYPE	co_ISA_SLTIU_SET_TO_1_abstract	COVERED	0.07	Bm
	26.	BASE ITYPE	co_ISA_SLTIU_SET_TO_1_abstract_JG	COVERED	0.91	Ht
	27.	BASE ITYPE	co_ISA_SLTIU_SET_TO_1	COVERED	0.07	Bm
	28.	BASE ITYPE	co_ISA_SLTIU_SET_TO_1_JG	COVERED	0.91	Ht
	29.	BASE ITYPE	co_ISA_SLTIU_SET_TO_0_abstract	COVERED	0.07	Bm
	30.	BASE ITYPE	co_ISA_SLTIU_SET_TO_0_abstract_JG	COVERED	0.95	Ht



Example reporting bugs

formalisx.	≡									
88 Overview	Asse	erts								
REPORT LISTS	No.	Instruction type	Property label	Assert status	Preconditions	Proof time	Engine	Bug	Mnemonic	Specifications
COVERS	1.	BASE ITYPE	as_ISA_ADDI_abstract	UNDETERMINED	COVERED	42783.41	Tri	Maybe	add rd rs1 imm12	x[rd] = x[rs1] + imm12. Adds imm12 to register x[rs1] and writes the result to x[rd], arithmetic overflow is ignored.
ASSUMPTIONS	2.	BASE ITYPE	as_ISA_ADDI	UNDETERMINED	COVERED	86063.33	Bm		add rd rs1 imm12	x[rd] = x[rs1] + imm12. Adds imm12 to register x[rs1] and writes the result to x[rd], arithmetic overflow is ignored.
	З.	BASE ITYPE	as_ISA_XORI_abstract	PROVEN	COVERED	90.19	М	NO	xori rd rs1 imm12	$x[rd] = x[rs1] \land Imm12$. Computes the bitwise XOR of registers $x[rs1]$ and Imm12 and writes the result to $x[rd]$.
	4.	BASE ITYPE	as_ISA_XORI	PROVEN	COVERED	73.07	М	No	xori rd rs1 imm12	$x[rd] = x[rs1] \land imm12$. Computes the bitwise XOR of registers $x[rs1]$ and imm12 and writes the result to $x[rd]$.
	5.	BASE ITYPE	as_ISA_ORI_abstract	PROVEN	COVERED	67.32	М	No	ori rd rs1 imm12	x[rd] = x[rs1] imm12. Computes the bitwise OR of registers x[rs1] and imm12 and writes the result to x[rd].
	6.	BASE ITYPE	as_ISA_ORI	PROVEN	COVERED	86.84	м	NO	ori rd rs1 imm12	x[rd] = x[rs1] imm12. Computes the bitwise OR of registers x[rs1] and imm12 and writes the result to x[rd].
	7.	BASE ITYPE	as_ISA_ANDI_abstract	PROVEN	COVERED	99.02	М	NO	andi rd rs1 imm12	x[rd] = x[rs1] & imm12. Computes the bitwise OR of registers x[rs1] and imm12 and writes the result to x[rd].
	8.	BASE ITYPE	as_ISA_ANDI	PROVEN	COVERED	66.73	м	ND	andi rd rs1 imm12	x[rd] = x[rs1] & imm12. Computes the bitwise OR of registers $x[rs1]$ and imm12 and writes the result to $x[rd]$.
	9.	BASE ITYPE	as_ISA_SLTI_SET_TO_1_ab stract	PROVEN	COVERED	79.05	Tri	No	slti rd rs1 imm12	x[rd] = x[rs1] <s and="" compares="" imm12.="" x[rs1]="" x[rs2]<br="">as signed numbers, and writes 1 to x[rd] if x[rs1] is smaller, and 0 if not.</s>
	10.	BASE ITYPE	as_ISA_SLTI_SET_TO_1	PROVEN	COVERED	35.67	Tri	No	slti rd rs1 imm12	x[rd] = x[rs1] <s and="" compares="" imm12.="" x[rs1]="" x[rs2]<br="">as signed numbers, and writes 1 to x[rd] if x[rs1] is smaller, and 0 if not.</s>

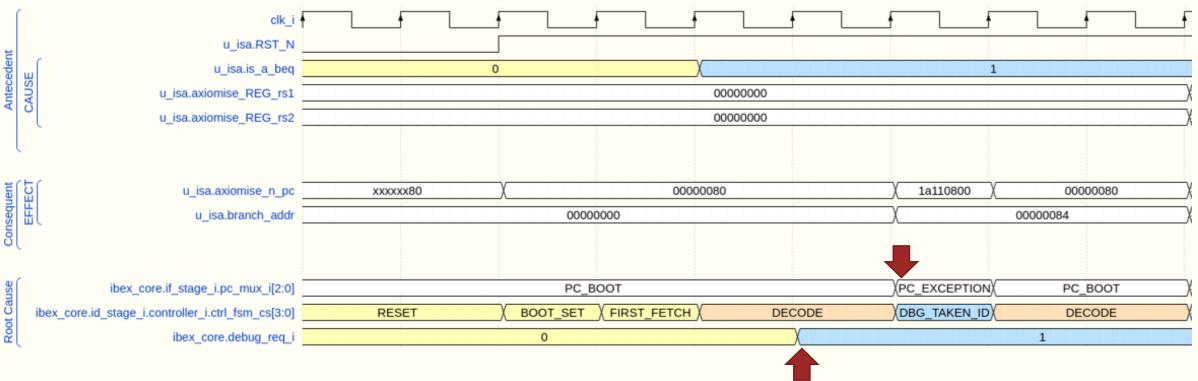


Anatomy of bugs Processor bugs caught by *formalISA*



BEQ Failure

Functional verification - ibex



Bug caused due to incoming debug request on the debug interface when the controller is in the DECODE state.

Nothing in the design to take care of such requests, causing the PC to be not updated correctly.

Root Cause

BEQ Failure

Functional verification - ibex

Only seen when debug arrives and the controller FSM is in the DECODE state.

Precise timing of arrival of debug makes this bug really hard to catch in dynamic simulation.

Formal catches it in seconds in 7 cycles!



Communication on ibex

Corner-case bugs confirmed

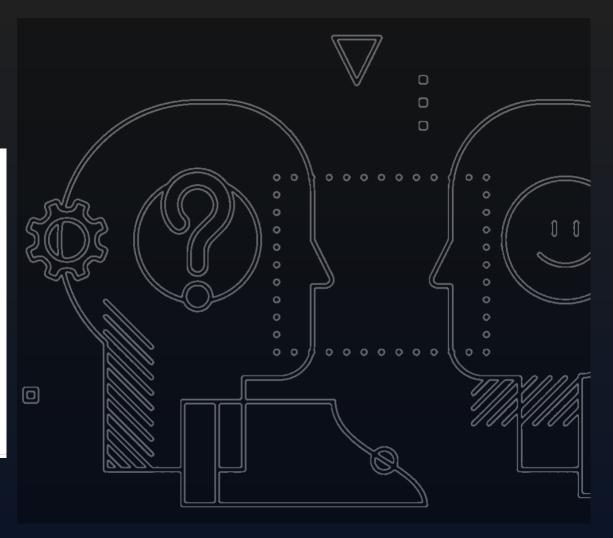
Hi Philip

Further, on my analysis on enabling full debug to see which arch checks pass/fail, it appears that the extent of damage goes beyond the six instructions I initially reported. Actually, other than SLLI, SRAI, and SRLI, all the checks fail.

The pattern is similar to what I reported in the six VCDs that I already assigned to you. If you want I can send you the tarball but don't want to overwhelm you. We're talking about investigating 51 additional property failures besides the six initially reported on branch instructions.

Instructions that fail are:

- BEQ, BGES, BLTS, BLTU, BNE (all reported before)
- ADDI, ADD, AND, ANDI, AUIPC, LUI, JAL, JALR, OR, ORI, XOR, XORI, SUB, SLTUI, STUI, SRL, SLL, SLTI, SLTS, SLTSI
- All checks that establish correctness for different variants (BYTE, WORD, HALF WORD) of LOAD/STORES (aligned/misaligned)





BEQ not working as expected #2 darbaria opened this issue on 26 Jun · 3 comments



imphil commented 2 hours ago

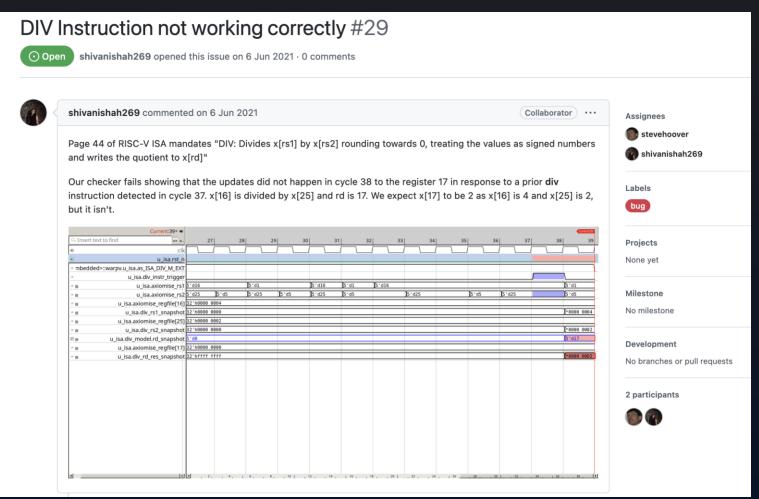
+ 😐 🚥



Hi @darbaria we have recently reworked the controller in this area and I'd expect this bug to be fixed. Can you test if it is still present?

WARP-V

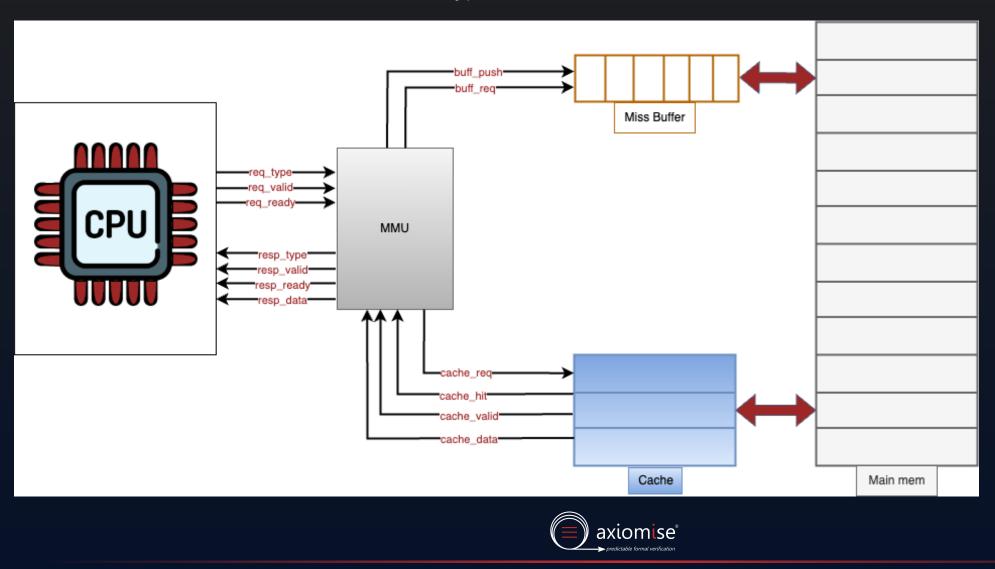
Six stage pipelined processor with a range of bugs

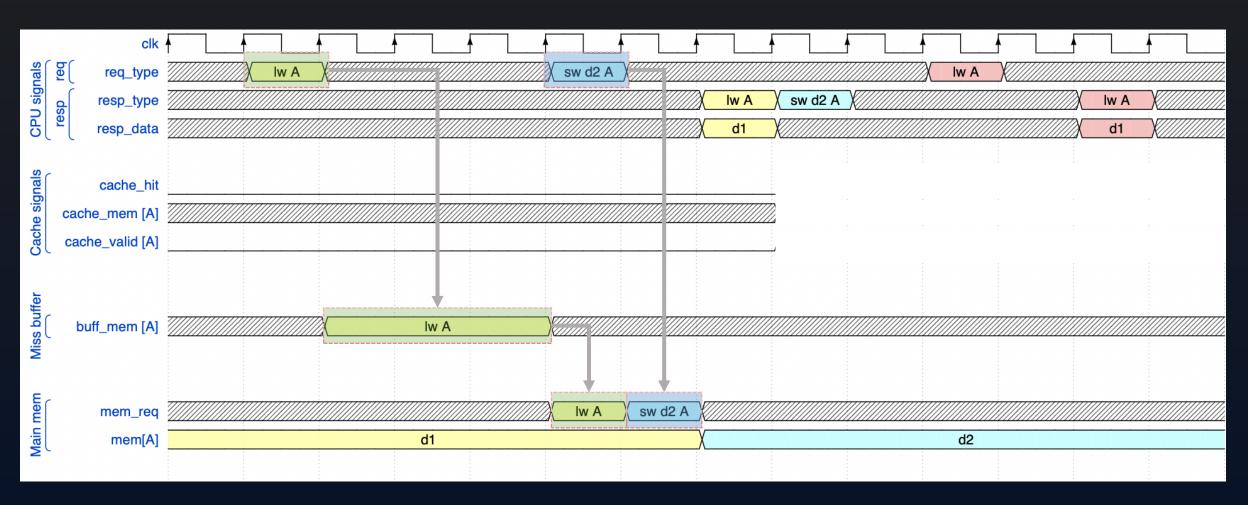




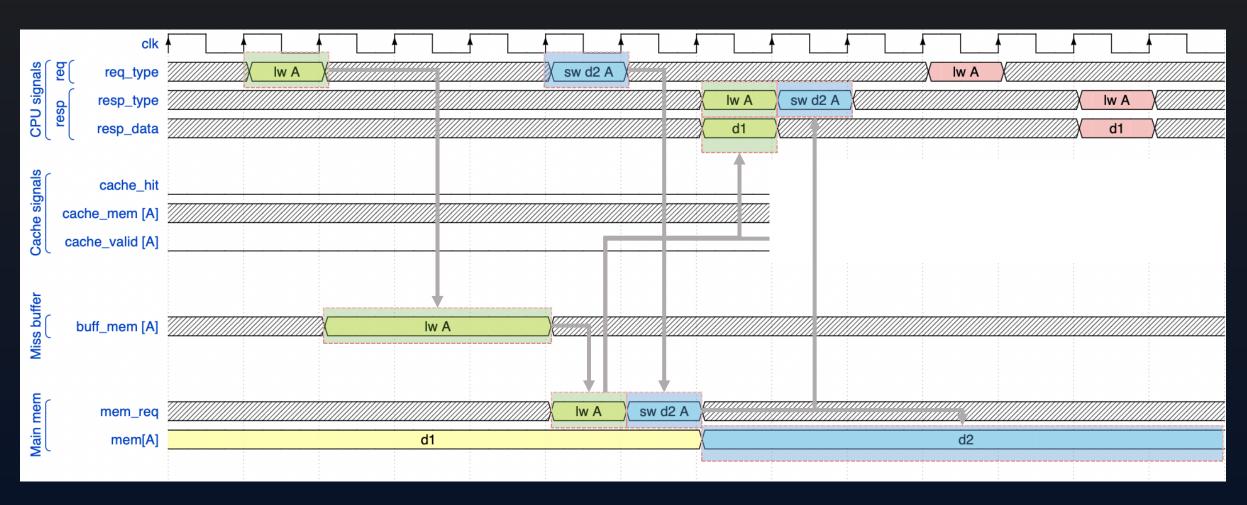
Memory subsystem Caught by our *formalISA*



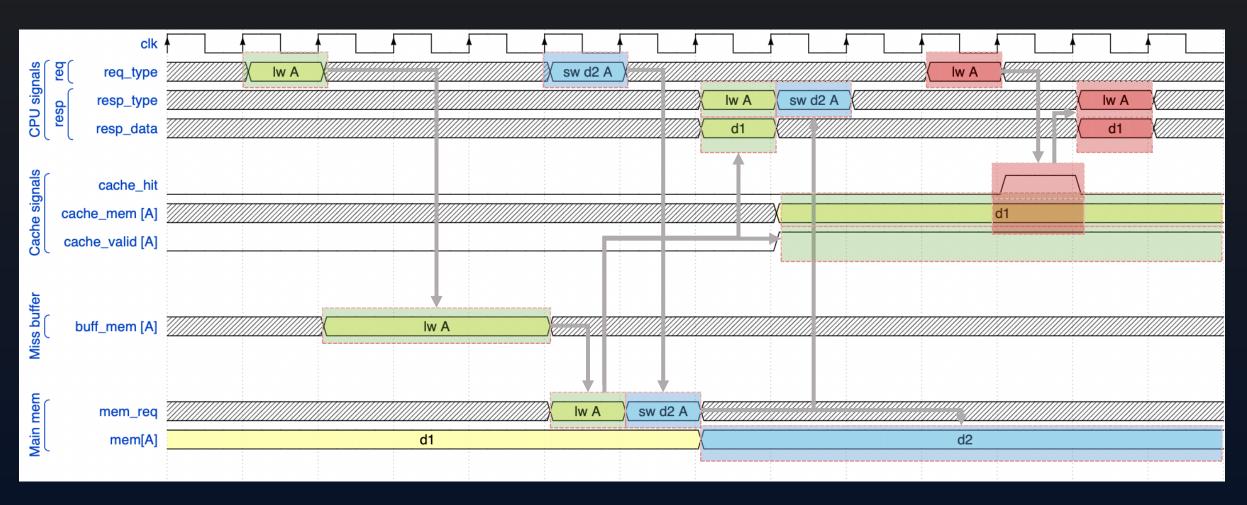














CHERIOT-IBEX

Corner case bugs



Illegal instruction handling

Verified in September 2024

The illegal instruction affected the execution of the valid instruction that followed it.

Ð	clk_i			
^л	u_ibex_core.id_stage_i.decoder_i <mark>opcode</mark>	0PCODE_AUIPC 0PCODE_LOAD 0PCODE_OP	OPCODE_LUI	
л	u_ibex_core.id_stage_i.decoder_i <mark>illegal_reg_cher</mark> i			
л	u_ibex_core.wb_stage_i			
л	u_ibex_core.wb_stage_i			
л	u_ibex_core.id_stage_i.instr_executing			
	data req o			
л	u_ibex_core.id_stage_i instr_executing			
л 🕀	u_isa.axiomise_regfile[10]		32'h5a	5a96_9e81
£ 🕂	gen_regfile_cheriot.register_file_i.rf_reg_q[10]		32'h5a	5a96_9e81

Issues

- Sending the illegal instruction request to the memory.
- Wasted execution power.
- Invalid data in the register file and subsequently in memory.

The illegal load instruction affected the execution of the valid AND (or any R-TYPE) instruction that followed it.



Illegal instruction handling – bit manipulation

After the first bug fix, bit manipulations instructions were broken

Current:16▼			Current:16	kliuMsft on Oct 15, 2024 Contributor ···
Q→Inse Reset	2 4 6	8 10 12 1	4 16 18 20	
-€ clk_i				Looking further into the issue, the culprit seems to be that the id_fsm_d logic can't handle exception
ா (_top.u_isa.BITMANIP_ZBS_as_ISA_BINVI				being issued in the 2nd half of a multi-cycle instruction. Specifically, the illegal_reg_cheri results in
л u_isa.zbs_binvi_instr_trigger				an EX stage exception but instr_kill is only raised in the 2nd half of a bit manipulation instruction
[™]	5'h00	5'h01	5'h06 5'h00 5'h18	(when rs3 is accessed). In this case multicycle_done is never issued and thus id_fsm_q will not
	32'h0000_0000		*004_0000	updated properly.
^л ⊞ u_ibex_core.id_stage_i.alu_operand_a	32'h0000_0000		*0004_0000_*000_0000	
^л ⊞ u ibex core.id stage i.alu operand b	32'h0000_0000		*0000_068e *0000_0000 *000_0480	@GregAC do you plan to keep supporting the bit instructions with rs3? if so I can try fix the behavior
	32'h0000_0000		*000 *000 *000 0000	in cheriot-ibex. You may want to take a look at the upstream ibex implementation as well.
	32'h0000_0000		*0000_068e_*000_0000 *000_0480	
^л ⊞ ∗x_core.ex_block_i.alu_i.shift_amt[4:0]	5 'h00		5'h12 5'h0	
		OPCODE_OP	*DE_OP_IMM *DE_RANCH 7'h43	
B e.id_stage_i.decoder_i.alu_operator_o		ALU_FSL	ALU_BINV ALU_E ALU_SLTU	
ⁿ u_ibex_core.id_stage_i.instr_executing				
u_ibex_core.id_stage_i.id_fsm_q	FIRST_CYCLE		FIRS CYCLE	
🖶 (_core.ex_block_i.alu_i.instr_first_cycle_i				
ⁿ re.id stage i.decoder i.illegal reg cheri				

https://github.com/microsoft/cheriot-ibex/issues/51



Decoder Issues

Very interesting set of issues

Observed Behavior https://github.com/microsoft/cheriot-ibex/issues/47

The RISC-V ISA for bit manipulation states that for the 32-bit implementations of the ISA, for the RORI instruction the bit 25 needs to be 1'b0. However, in the implementation we have downloaded from the Github on 17 July 2024, an instruction could be considered to be a valid RORI despite the bit 25 being 1'b1. It means there are two ways to decode the same instruction which means it is prone to security vulnerabilities and it does not comply with the ISA.

clk_i					L				1		
u_ibex_core.id_stage_i.instr_rdata_i	32'h0000_000	9		32'h638	5_d193			32'h800	8_118f	32'h006	8_327b
u_ibex_core.id_stage_i.instr_rdata_i[25]	25]					<u> </u>					
u_ibex_core.id_stage_i.instr_executing											
u_ibex_core.if_stage_i.instr_fetch_err_o											
u_ibex_core.if_stage_i.instr_new_id_o		1									
u_ibex_core.id_stage_i.decoder_i.opcode	7'h00			OPCODE	OP_IMM			OPCODE	MISC_MEM	OPCODE	AUICGP
u_ibex_core.ex_block_i.alu_i.operator_i	ALU_SLTU			ALU_ROR				ALU SLT	Ú		
u_ibex_core.id_stage_i.decoder_i.instr[13:12]	2'600			2'b01						2'b11	
u_ibex_core.id_stage_i.decoder_i.instr[6:0]	7 'h00			7'h13				7'h0f		7 'h7b	
		1									1

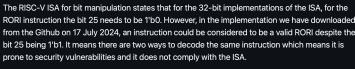


Decoder Issues

Very interesting set of issues



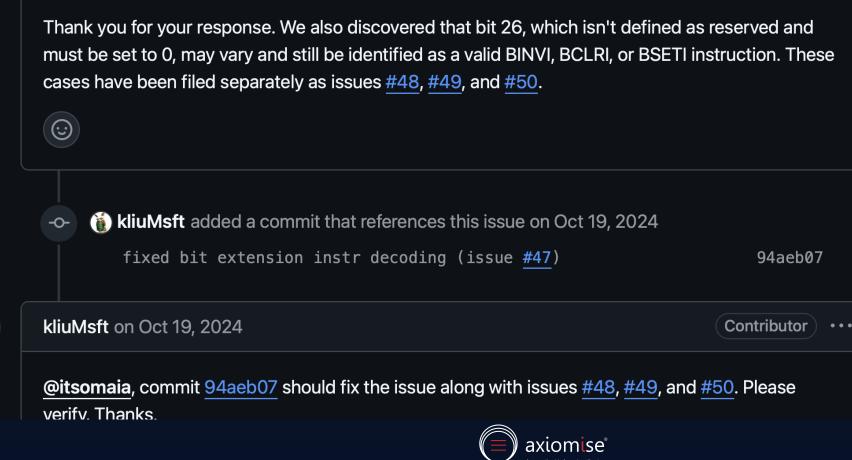
itsomaia on Oct 15, 2024 https://github.com/microsoft/cheriot-ibex/issues/47



Observed Behavior

Autho

clk_i			
u_ibex_core.id_stage_i.instr_rdata_i	32'h0000 0000	32'h6385_d193	32'h8008_118f 32'h0068_327b
u_ibex_core.id_stage_i.instr_rdata_i[25]			
u_ibex_core.id_stage_i.instr_executing			
u_ibex_core.if_stage_i.instr_fetch_err_o			
u_ibex_core.if_stage_i.instr_new_id_o			
u_ibex_core.id_stage_i.decoder_i.opcode	7'109	OPCODE_OP_IMM	OPCODE_MISC_MEN_OPCODE_AUICGP
u_ibex_core.ex_block_i.alu_i.operator_i	ALU_SLTU	ALU_ROR	ALU_SLTÚ
u_ibex_core.id_stage_i.decoder_i.instr[13:12]	21609	2'601	2'b11
u_ibex_core.id_stage_i.decoder_i.instr[6:0]	7*608	7'h13	7'h9f 7'h7b





Area saved

Area Analyser Daxiomis=? Esotiprint

Redundancy report

Design in



Footprint – Area analyser for silicon

Cheriot-ibex

Design gate count	Design flop count	Redundant components	Estimated redundant gates per category
303,737 14	14,723	Counter: 3	Counter:768
		Register: 313	Register:16,440
		Array: 23	Array:7872





Summary

Formal methods is a necessity not a nice-to-have

Bugs are a natural consequence of implementing design

Bugs caught late in the design cycle result in very expensive fixes and catastrophic failures

Formal == efficient bug hunting & exhaustive proofs right at the time of design bring up

Architectural validation must employ formal verification to build "proofs" of bug absence

Architects, designers, verification engineers can all use *formalisa* without any FV experience

Find bugs, build proofs, obtain inter-operable coverage model for use in simulation and other formal tools

Use any formal verification tool of your choice

Find corner-case bugs as well as build exhaustive proofs







www.axiomise.com CONSULTING & SERVICES TRAINING CUSTOM SOLUTIONS info@axiomise.com