UnityChip Verification: Scaling Out Hardware Verification with Software Testing Developers

Yunlong Xie, Zhicheng Yao, Sa Wang, Yungang Bao

Background

50%-60%
Median project time spent in verification

51%
Design engineer's time spent on design task

49%
Design engineer's time spent on verification task

Chip verification occupies a significant portion of project resources.

50%
Design engineers increase since 2007

146%
Verification engineers increase since 2007

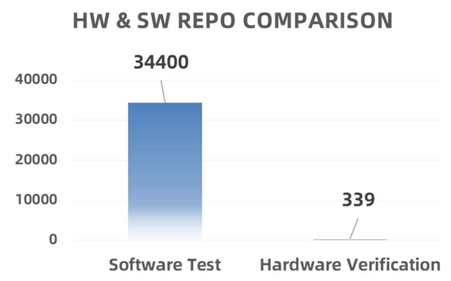
"Perhaps one of the biggest challenges today is to control cost and engineering headcount." [1]

[1]: 2022 Wilson Research Group Functional Verification Study. SIEMENS

Motivation

Open-source Verification Tools are Now Available

Success Stories of Community-driven Software Test



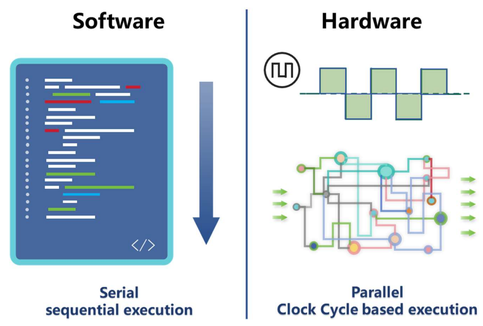
Utilize the Diverse Open-source Software to Enhance Hardware Ecosystem

It's time for open-source crowdsourced verification

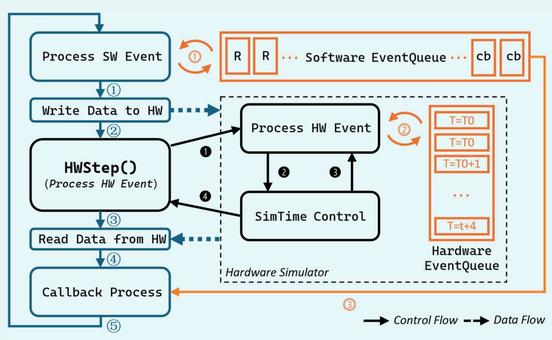
[2]: Life Post Moore's Law: The New CAD Frontier, Prof. Mark Horowitz, Stanford University

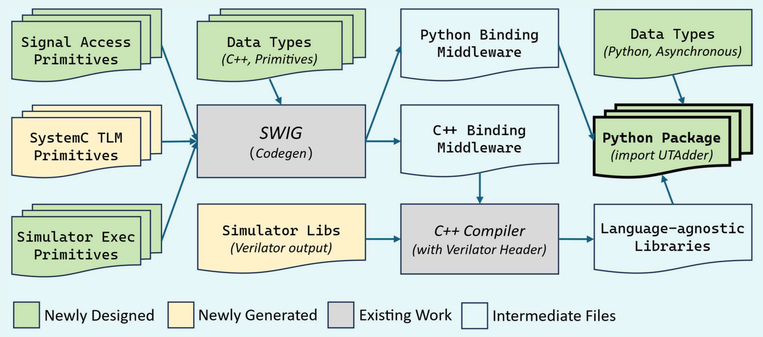
Challenges & Solutions

CHALLENGE 1:
Programming Paradigm Difference

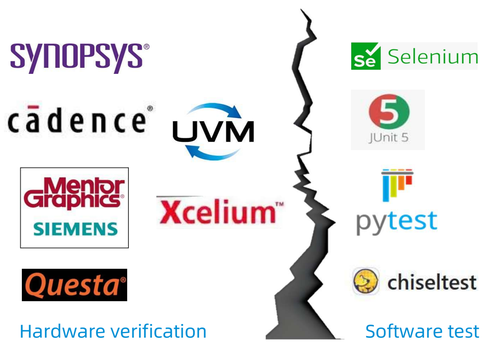


SOLUTION 1 : Multi-language Event-Driven Verification

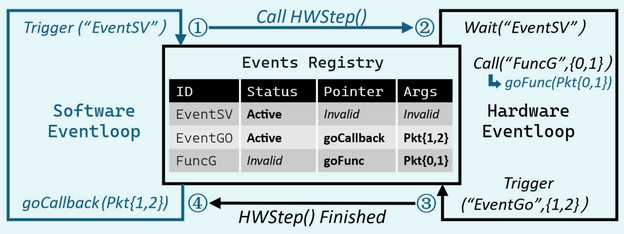


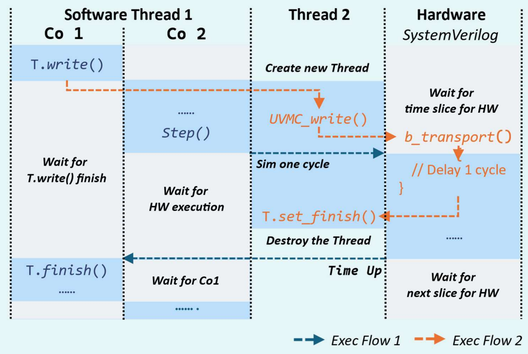


CHALLENGE 2:
Hardware/Software Cooperation

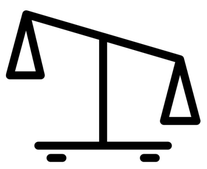


SOLUTION 2 : Transaction & Event Synchronization



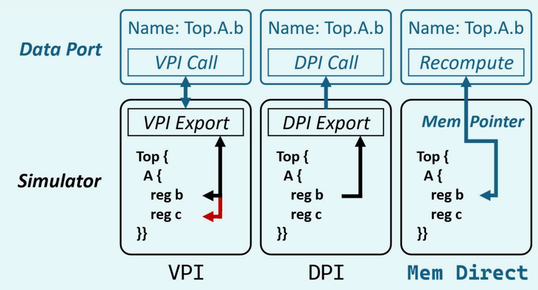


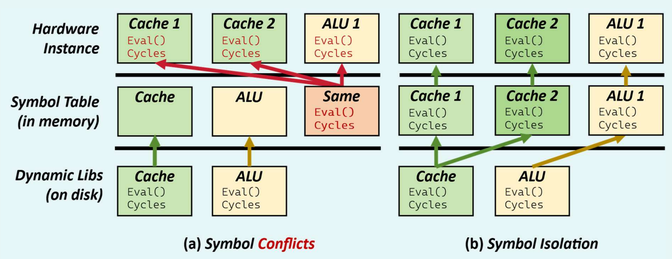
CHALLENGE 3:
Performance Debuggability Tradeoff



Open-source hardware tool chains trade off performance for functionality in their design. For example, even the state-of-the-art simulator Verilator faces conflicts between simulator optimization and Verilog Procedural Interface.

SOLUTION 3 : Software-Defined Optimization





Case Study

CS No.	Module	Tools	Participants					Documentation Support				Result	
			BG	Skill	Local	Num.	Participated	Tools	Design	Verification	Ref	Duration	Bugs
1	BPU	UVM	HW	Experienced	Onsite	1	Yes	Normal	Brief	None	None	5 months	2
2			SW	Junior		1							0
3		UCV				5	No					2 months	10
4					Remote	4							6
5	NoC	UVM	HW	Experienced	Onsite	2	Yes	Normal	Normal	Provided	Provided	4 months	/
6		UCV+	SW	Junior		1	No					2 weeks*	
7	Router	UCV	SW	Novice	Onsite	1	No	Detailed	Detailed	Provided	Provided	2 weeks	0
8	RAS					1					None		0
9	Decoder	UCV	HW	Experienced	Onsite	2	Yes	Normal	Normal	Provided	Provided	1 months	12
10	L2Cache	UCV	SW	Junior	Onsite	1	Yes	Normal	Normal	Provided	None	1 months	/

Participated: have any prior experience in verification work or not. Bugs: / indicates that this case is not aimed at finding bugs.
UCV+: a UVM environment enhanced by UCV. The 2 weeks* is due to CS#6 being optimized based on CS#5.

Module	Design LOC	Tools	Verify LOC	Compilation	Execution
NoC	13,036	**UCV+**	**13,766**	24.06s	**15.41s**
		UVM	15,664	15.32s	18.47s
ICache	5,163	**UCV+**	**7,211**	**13.69s**	**94.36s**
		UVM	8,063	19.14s	106.12s

UCV+ is a UVM environment enhanced by UCV.

- **Software languages help, but the lower barrier is the key.**
- **Documentation matters.**
- **Participants' interest and enthusiasm take precedence over technical expertise in verification results**

Results

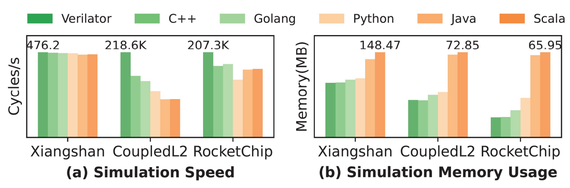


Fig 1. Event-Driven Performance

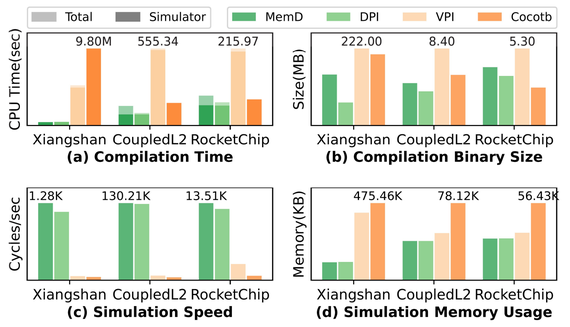


Fig 2. Performance with Debuggability

Inspired by the open-source hardware ecosystem and the benefits of software communities, we propose a multi-aspect optimization approach for software-based hardware verification, the UnityChip Verification platform.

Evaluated on the XiangShan and Rocket-Chip RISC-V processor, our framework achieves up to **20× runtime speedup and 76% memory savings** over Cocotb, offers zero-overhead support for C++, Python, and other languages, and **reduces code by 12% and accelerates speed by 16.6%** while reusing VIPs.

This platform establishes a hardware verification toolchain that enables software engineers to verify chips more efficiently.