RETROSPECTIVE: FireSim: FPGA-Accelerated Cycle-Exact Scale-Out System Simulation in the Public Cloud

Sagar Karandikar[1], Howard Mao[2,*], Donggyu Kim[3,*], David Biancolin[4,*], Alon Amid[5,*], Dayeol Lee[6,*], Nathan Pemberton[7,*], Emmanuel Amaro[8,*], Colin Schmidt[4,*], Aditya Chopra[2,*], Qijing Huang[5,*], Kyle Kovacs[9,*], Borivoje Nikolić[1], Randy Katz[1], Jonathan Bachrach[10,*], Krste Asanović[1]

[1]UC Berkeley, [2]Google, [3]Apple, [4]SiFive, [5]NVIDIA, [6]Anyscale, [7]Amazon Web Services, [8]VMWare Research, [9]DiCon Fiberoptics, [10]JITX

I. Introduction

"Why is it called *FireSim*?" is a question we receive often, posed by users who today employ FireSim to simulate a variety of systems beyond its initial goal: as a "from-the-RTL-up" simulator for a specialized Warehouse-Scale Computer (WSC) architecture called *FireBox* [O8][1]. When we set out to build FireSim, the published mandate [O8] was to:

> *"...simulate an entire FireBox, including the fiber-optic network, the switch, the NIC, and 1000 SoCs, with every core running the full BDAS stack (from the AMP Lab) and the Linux OS, as well as interactive services and batch applications, with only a factor of 1000x slowdown from realtime."*

The FireSim ISCA'18 paper [13] exceeded these objectives, with one caveat: our demo applications were not JVM-based as no usable RISC-V JVM existed in 2017. While the achieved scale was exciting, the true promise and broader adoption of FireSim has been driven by *how* this scale was realized.

II. Prior FPGA-accelerated Simulation Efforts

Prior efforts used FPGAs for HW modeling, including at WSC scale (see related work in [13]). While these simulators pushed forward in capability, they still faced technical hurdles that prevented wide adoption. Many were built from abstract and hand-written RTL models, requiring users to be *simulation* experts and hand-design *RTL simulators* of their ideas, a much greater effort than writing tapeout-friendly RTL. New models would then require validation, so in novel domains, a tapeout-friendly RTL baseline would still require development. Many of the simulators also used extremely expensive and/or custom FPGA platforms. Reproducing the host platforms themselves was difficult, hampering end-to-end experiment reproduction.

III. Hardware Trends in 2017

Around 2017, when building FireSim, we identified several technology trends that helped us overcome these issues:

FPGAs in the public cloud became broadly available [5]. Cloud computing, established for years in systems research [O23], could now benefit architects. Academics/startups could elastically scale high-fidelity simulations to 1000s of nodes without buying millions of dollars of FPGAs. In large organizations, architects/systems SW developers, who rarely get access to costly "big metal" HW-accelerated simulators, could now co-design HW/SW directly using real RTL.

FPGA capacity grew sufficiently to host interesting research targets without *immediately* requiring "tricks" (multi-threading, abstract modeling, partitioning, etc.) from the FPGA simulator literature. Many were later added to FireSim, but critically, were not initially *required* to ship a useful simulator.

Open-source, industry-verified hardware implementations became available. These were sufficiently capable to serve as a base for architecture research and included microprocessors, caches, on-chip networks, and peripherals [1]–[3].

RISC-V brought broad SW support for open HW designs, allowing them to run entire operating systems and applications.

Intermediate representations of RTL enabled compiler passes that automatically transformed HW designs [10], [O27].

IV. FireSim's Design Philosophy

Several guiding principles enabled FireSim to scale from high-fidelity simulations of single SoCs to entire WSCs:

Treat FPGAs as a "simulation appliance." Users should focus on their design task, not FPGA platform specifics. FireSim took a cloud-first approach, with heavy automation to hide the complexities of using FPGAs and ample documentation (https://docs.fires.im). This automation was *required* to enable the simulation scale presented in the original paper, but all users today, including those simulating single systems or using on-premises FPGAs, reap the benefits.

Parameterized tapeout-friendly RTL should be the single source of truth. The vast majority of users should not have to modify and re-calibrate abstracted models.

FPGA simulation "tricks" should be applied by a compiler, e.g., FireSim's multi-ported RAM [18] and multi-threading optimizations [7]. Abstract models should be used

[*] All authors were affiliated with UC Berkeley when this work was completed. Current affiliations are provided only for reference.

[1] Citations of the form [O..] refer to those in the original paper.

only when necessary and must be heavily validated and reused (e.g., FireSim's FASED DRAM model [6]).

One flow should scale from high-level architectural modeling and software implementation to pre-silicon verification/validation for tape-out. This required coherently packaging everything from RTL designs and VLSI tools to compatible software [19]. In 2019, as the flow grew, parts were moved to a more logical home in Chipyard [4].

Easy-to-use instrumentation and deterministic simulation are essential. FirePerf [14] and DESSERT [15] added high-fidelity and deterministic debugging and performance instrumentation to FireSim, enabling SW-simulator-like flexibility and introspection, but at significantly higher speeds.

V. Open-source and the FireSim community

FireSim was open-sourced in May 2018 alongside the publication of the ISCA paper (github.com/firesim/firesim). Since then, FireSim has matured into the de facto open-source high-performance FPGA-accelerated simulation platform for designs at various scales. Many tutorials at ISCA, MICRO, ASPLOS, and HPCA have given hundreds of attendees hands-on experience running FireSim simulations on cloud FPGAs. The first FireSim *Workshop*, co-located with ASPLOS 2023, brought together the FireSim community with a day of talks from external FireSim users (fires.im/workshop-2023).

FireSim has been *used* in numerous publications at top conferences across various EE/CS-focused research domains, including computer architecture, systems, networking, security, scientific computing, circuits, design automation, and more. This validates our vision that an easy-to-use, high-performance simulator of realistic RTL hardware implementations would provide a shared platform to enable HW/SW co-design and collaboration between researchers at varying levels of the computing stack. The complete list of FireSim user publications and user institutions is too long to list here; see the FireSim website: fires.im/publications/#userpapers.

Several users have deployed FireSim in surprising ways. For example, FireSim was used as a host platform for DARPA's first ever bug bounty program, FETT, to make several novel security-augmented hardware designs available to hundreds of white-hat hackers for security evaluation over the internet [16]. Industrial users have also published comparisons of FireSim simulations of their designs against their taped-out silicon, providing an end-to-end validation of FireSim's performance modeling capabilities [17]. Other users have even deployed FireSim for its *original* purpose of modeling novel, *scale-out* systems [9], [11]. In addition to its original Chisel HDL support, FireSim has also been used with Verilog designs, such as those generated by HLS toolchains [8].

FireSim was designed from the ground-up to support reproducibility, which has been a challenge in architecture research. The initial FireSim release included a script that reproduced the experiments from the ISCA paper, including the largest scale-out simulations. Many FireSim user papers have since undergone artifact evaluation processes now part of conferences, including receiving distinguished artifact awards [12].

Today, FireSim is actively developed by a global group of contributors. The latest FireSim release, coinciding with our tutorial at ISCA-50, supports several *on-premises* FPGA boards, including desktop/server-class boards requested by users (Xilinx U250, U280, and VCU118). Also added is the RHS Research Nitefury II, an exciting low-cost, portable board that works with a laptop via Thunderbolt or M.2. FireSim's on-premises FPGA support was added with the aforementioned design principles in-mind, maintaining the automation and abstraction that have made cloud-hosted FireSim a powerful tool. Users can also easily transition between on-premises and cloud FPGAs, enabling a *hybrid-cloud* approach where early development occurs on a small cluster of on-premises FPGAs, with the ability to burst to cloud FPGAs during deadlines.

Looking forward, we are excited to see how FireSim and the broader open-hardware community evolve and are grateful for the opportunity to reflect on FireSim's impact thus far.

Acknowledgements

We thank the many additional contributors to FireSim since ISCA 2018, including: Abraham Gonzalez, Albert Magyar, Albert Ou, Tim Snyder, Nandor Licker, Jerry Zhao, Björn Gottschall, David Metz, Farzad Fatollahi-Fard, Sarah Zhou, Benjamin Morse, Tushar Sondhi, Russell Horvath, Jack Koenig, Farzad Farshchi, Joonho Whangbo, Kevin Anderson, Lux Zhang, Edwin Lim, Yakun Sophia Shao, and many other contributors/sponsors in [13] and follow-on works.

References

[1] https://github.com/chipsalliance/rocket-chip.

[2] https://github.com/sifive/block-inclusivecache-sifive.

[3] https://github.com/sifive/sifive-blocks.

[4] A. Amid *et al.*, "Chipyard: Integrated design, simulation, and implementation framework for custom SoCs," *IEEE Micro 40.4*, 2020.

[5] J. Barr, "EC2 F1 instances with FPGAs," https://aws.amazon.com/blogs/aws/ec2-f1-instances-with-fpgas-now-generally-available/, 2017.

[6] D. Biancolin *et al.*, "FASED: FPGA-accelerated simulation and evaluation of DRAM," in *FPGA 2019*.

[7] D. Biancolin *et al.*, "Accessible, FPGA resource-optimized simulation of multiclock systems in FireSim," *IEEE Micro 41.4*, 2021.

[8] Q. Huang *et al.*, "Centrifuge: Evaluating full-system HLS-generated heterogenous-accelerator SoCs using FPGA-acceleration," in *ICCAD'19*.

[9] S. Ibanez *et al.*, "The nanoPU: A nanosecond network stack for datacenters," in *OSDI 2021*.

[10] A. Izraelevitz *et al.*, "Reusability is firrtl ground: Hardware construction languages, compiler frameworks, and transformations," in *ICCAD'17*.

[11] T. Jepsen *et al.*, "From sand to flour: The next leap in granular computing with NanoSort," 2022.

[12] S. Karandikar *et al.*, "A hardware accelerator for protocol buffers," in *MICRO 2021*.

[13] S. Karandikar *et al.*, "FireSim: FPGA-accelerated cycle-exact scale-out system simulation in the public cloud," in *ISCA 2018*.

[14] S. Karandikar *et al.*, "FirePerf: FPGA-accelerated full-system hardware/software performance profiling and co-design," in *ASPLOS 2020*.

[15] D. Kim *et al.*, "DESSERT: Debugging RTL effectively with state snapshotting for error replays across trillions of cycles," in *FPL 2018*.

[16] J. Kiniry, "FireSim in high-profile action—FETT: DARPA's first ever bug bounty program," in *2023 FireSim Workshop*. [Online]. Available: https://fires.im/workshop-2023-pdf/02_Keynote_Kiniry.pdf

[17] Y. Lee and A. Waterman, "Managing chip design complexity in the domain-specific SoC era," in *VLSI 2020*.

[18] A. Magyar *et al.*, "Golden Gate: Bridging the resource-efficiency gap between ASICs and FPGA prototypes," in *ICCAD'19*.

[19] N. Pemberton and A. Amid, "FireMarshal: Making HW/SW co-design reproducible and reliable," in *ISPASS 2021*.