



Easy-to-use, FPGA-accelerated  
Cycle-accurate Hardware  
Simulation in the Cloud

<https://fires.im>

 @firesimproject

Sagar Karandikar, **Howard Mao**, David Biancolin, Alon Amid, Nathan Pemberton, Albert Ou  
Borivoje Nikolić, Randy Katz, Jonathan Bachrach, Krste Asanović



Berkeley **Architecture Research**



# Bridging The Gap Between Open HW and Arch. Research

- Why did architecture researchers traditionally write abstract SW simulators?
  - No access to reasonable RTL
  - Expensive FPGAs / Proprietary HW-accel simulation tools
  - FPGA Prototyping is not good enough for arch research
  - FPGAs hard to use—each platform is different
- RISC-V driving a huge explosion in open hardware + complex compatible software stacks
- ***FireSim bridges the gap between architecture researchers, who expect productive simulators, and the blossoming open-hardware movement***



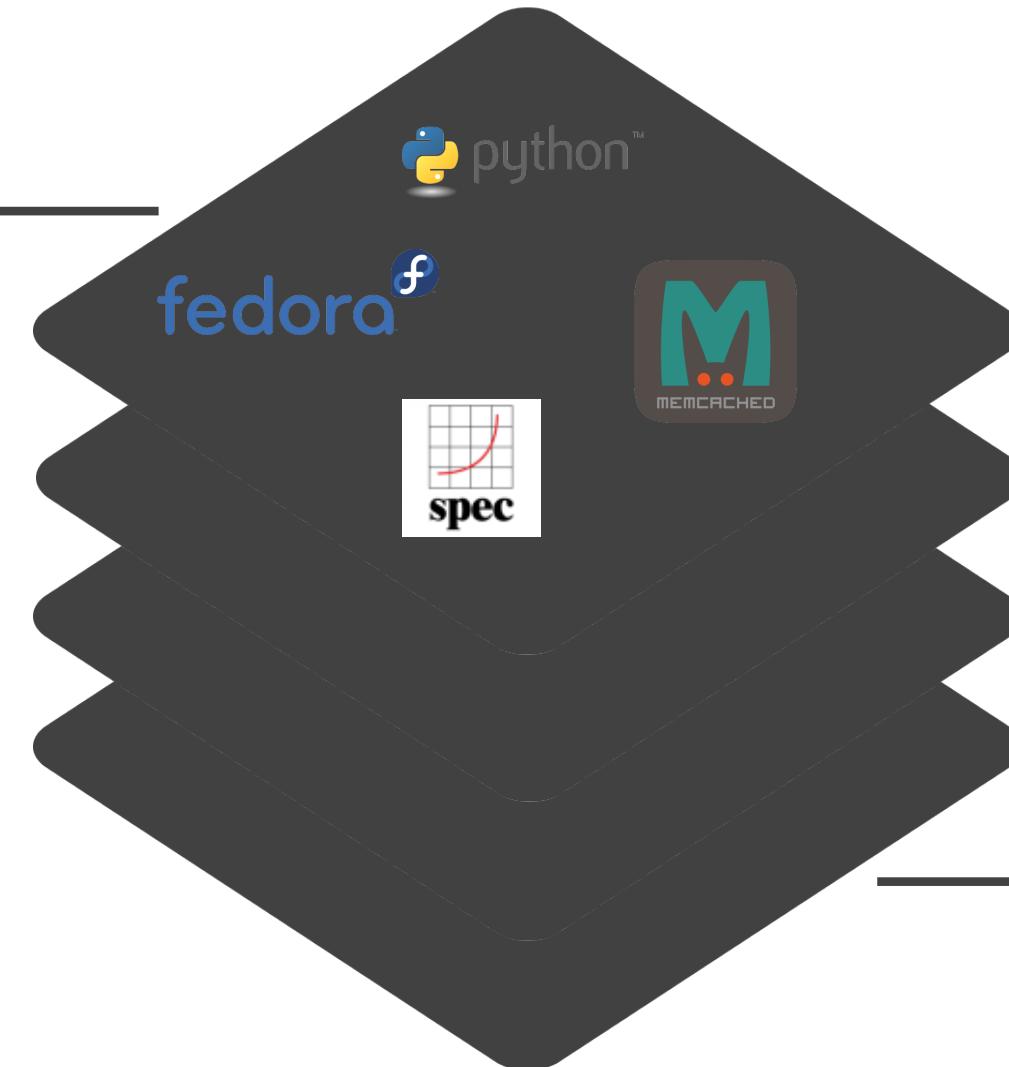
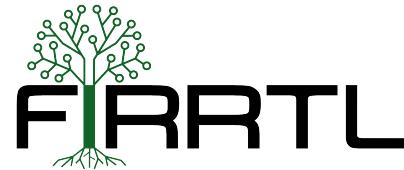


# Useful Trends Throughout the Stack

Open ISA



High-Productivity  
Hardware IR



Open, Silicon-Proven  
SoC Implementations



FPGAs in the Cloud



Amazon EC2 F1 Instances

Run Customizable FPGAs in the AWS Cloud





# Want:

- Faster than software simulator
- As detailed as silicon
- All the benefits of SW-based simulator
- Low cost

# Our Thesis:

- FPGAs are the only viable basis technology  
→ Build *FPGA-accelerated* simulators with SW-like features using *open-source* tools



# FireSim at 35,000 feet

- Open-source, fast, automatic, deterministic FPGA-accelerated hardware simulation for pre-silicon verification and performance validation
- Ingests:
  - Your RTL design (FIRRTL, either via Chisel or Verilog via Yosys\*)
    - Or included designs—Rocket Chip, BOOM, NVDLA, PicoRV32, and growing
  - HW and/or SW IO models (e.g. UART, Ethernet, DRAM, etc.)
  - SW workload descriptions
- Produces:
  - Fast, cycle-exact, scalable simulation of your design + models around it
  - Automatically deployed to cloud FPGAs (AWS EC2 F1)

S. Karandikar et. al., “FireSim: FPGA-Accelerated Cycle-Exact Scale-Out System Simulation in the Public Cloud.” *ISCA 2018*

S. Karandikar et. al., “FireSim: FPGA-Accelerated Cycle-Exact Scale-Out System Simulation in the Public Cloud.” *IEEE Micro Top Picks 2018*





# Three Distinguishing Features of FireSim

- 1) Not FPGA prototypes, rather FPGA-accelerated simulators
  - Akin to commercial co-simulation platforms
- 2) Uses cloud FPGAs
  - Inexpensive, elastic supply of large FPGAs
  - Easy to collaborate with other researchers
  - Heavy automation to hide FPGA complexity
- And of course...
- 3) Open-source (<https://fires.im>)





# Why is FPGA Prototyping Insufficient?

## Taped-out SoC Design

RTL  
taped-out  
1 GHz

DRAM  
100ns  
latency

SoC sees 100 cycle DRAM latency

## FPGA Prototyping

RTL  
on FPGA  
100 MHz

DRAM  
100ns  
latency

SoC sees 10 cycle DRAM latency





# The Difficulty with FPGA Prototypes

- Every FPGA clock executes one cycle of the simulated machine
- Exposes latencies of FPGA resources to the simulated world.

Three implications:

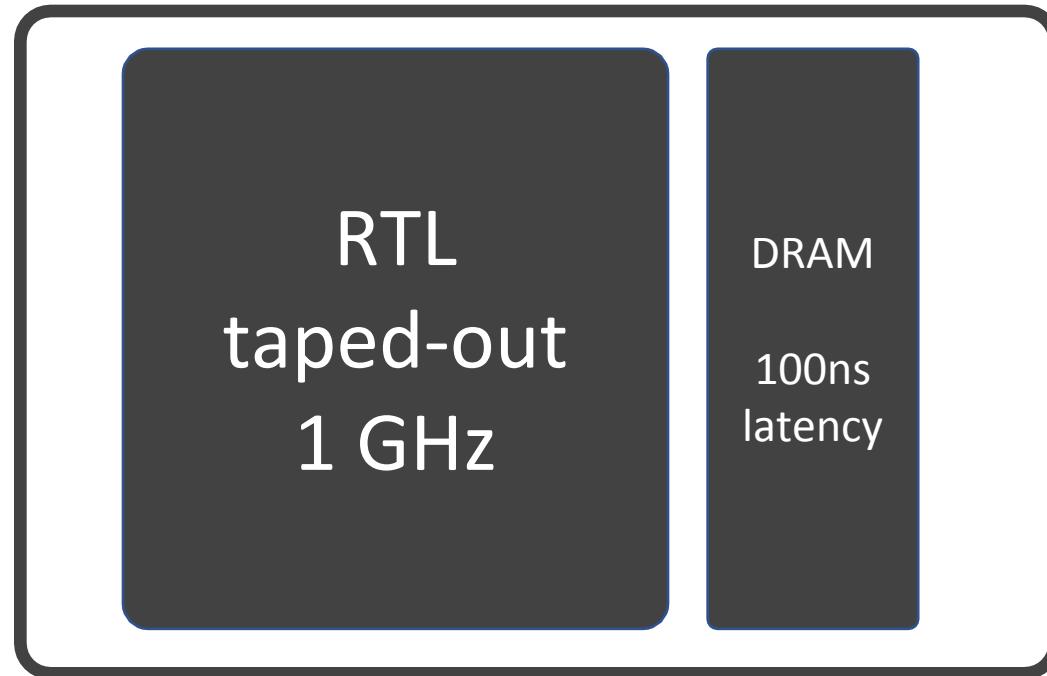
- 1) FPGA resources may not be an accurate model (ex. previous slide)
- 2) Simulations are non-deterministic
- 3) Different host FPGAs produce different simulation results





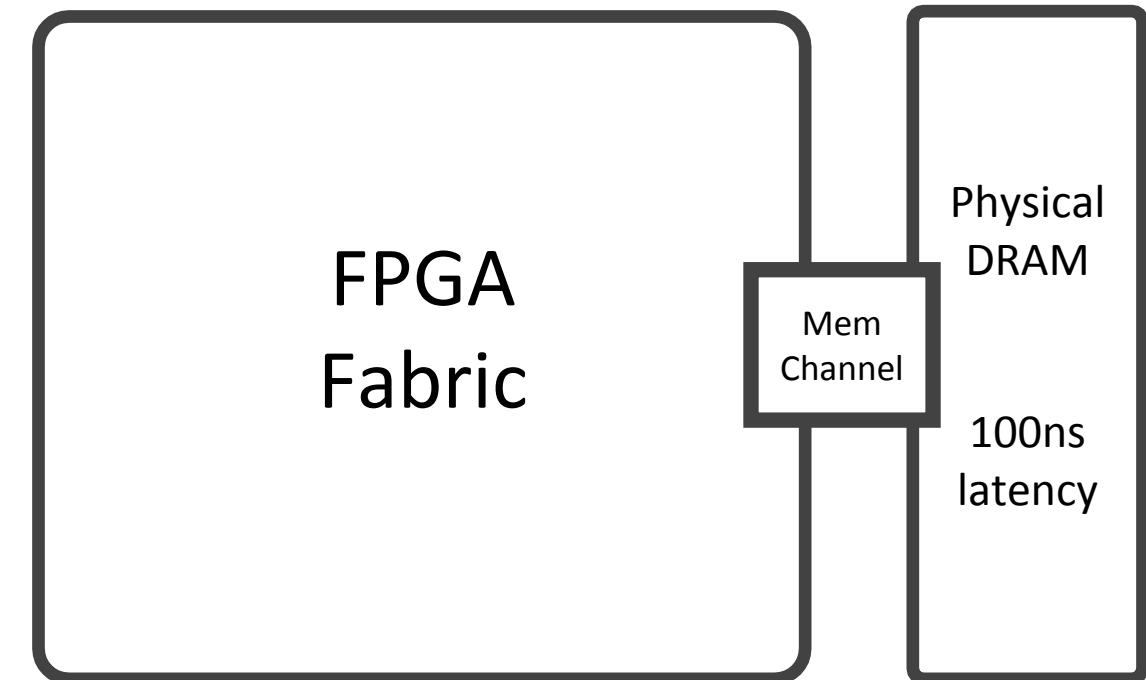
# Separating Target and Host

Target: the machine under simulation



*Closed simulation world.*

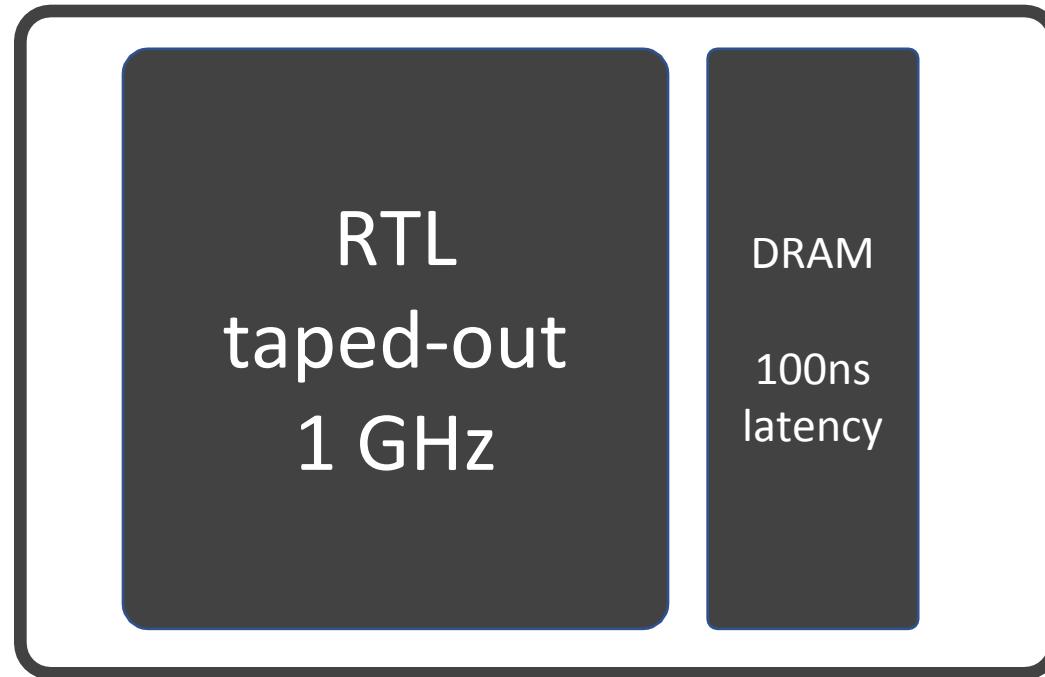
Host: the machine executing (*hosting*) the simulation





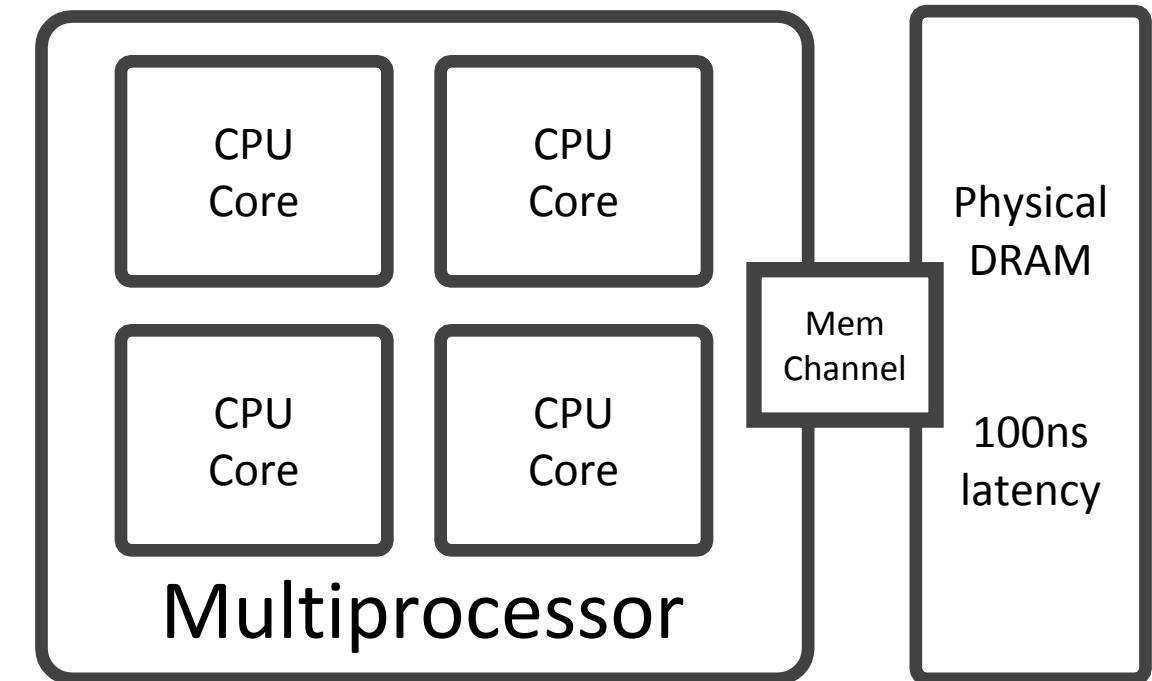
# Separating Target and Host

Target: the machine under simulation



*Closed simulation world.*

Host: the machine executing (*hosting*) the simulation





# Host Decoupling in FireSim: Transforming the Target

- 1) Convert RTL into a latency-insensitive [1] model using FIRRTL transform



- 2) Generate FPGA-hosted model for DRAM [2] (think DRAMSim on an FPGA)
- 3) Generate queues (token channels) to connect the target models

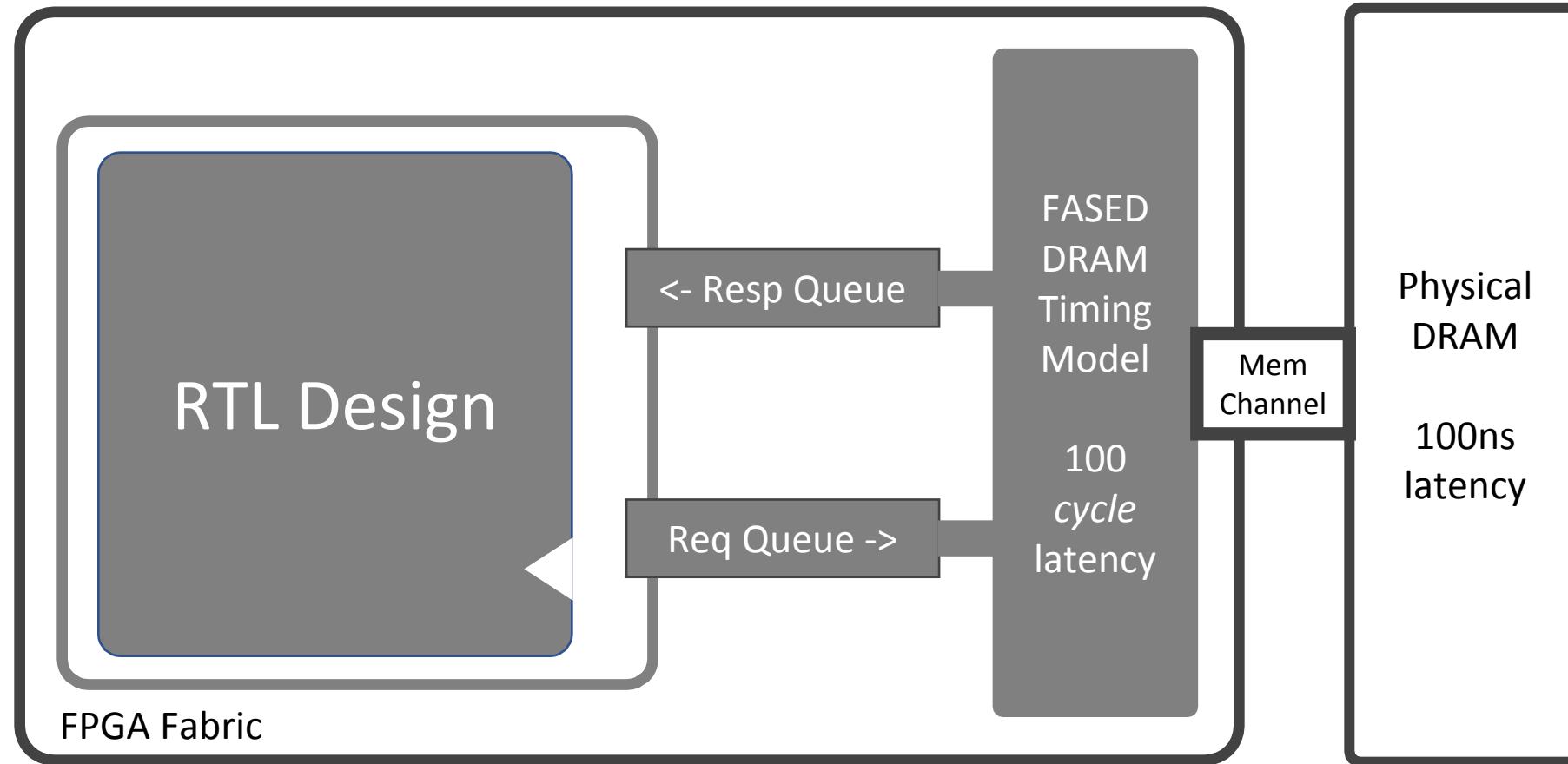
[1] *Theory of Latency Insensitive Design*, Carloni et al, also see: RAMP

[2] FASED: FPGA-accelerated Simulation and Evaluation of DRAM, Biancolin et al





# Host Decoupling in FireSim: Mapping to the FPGA

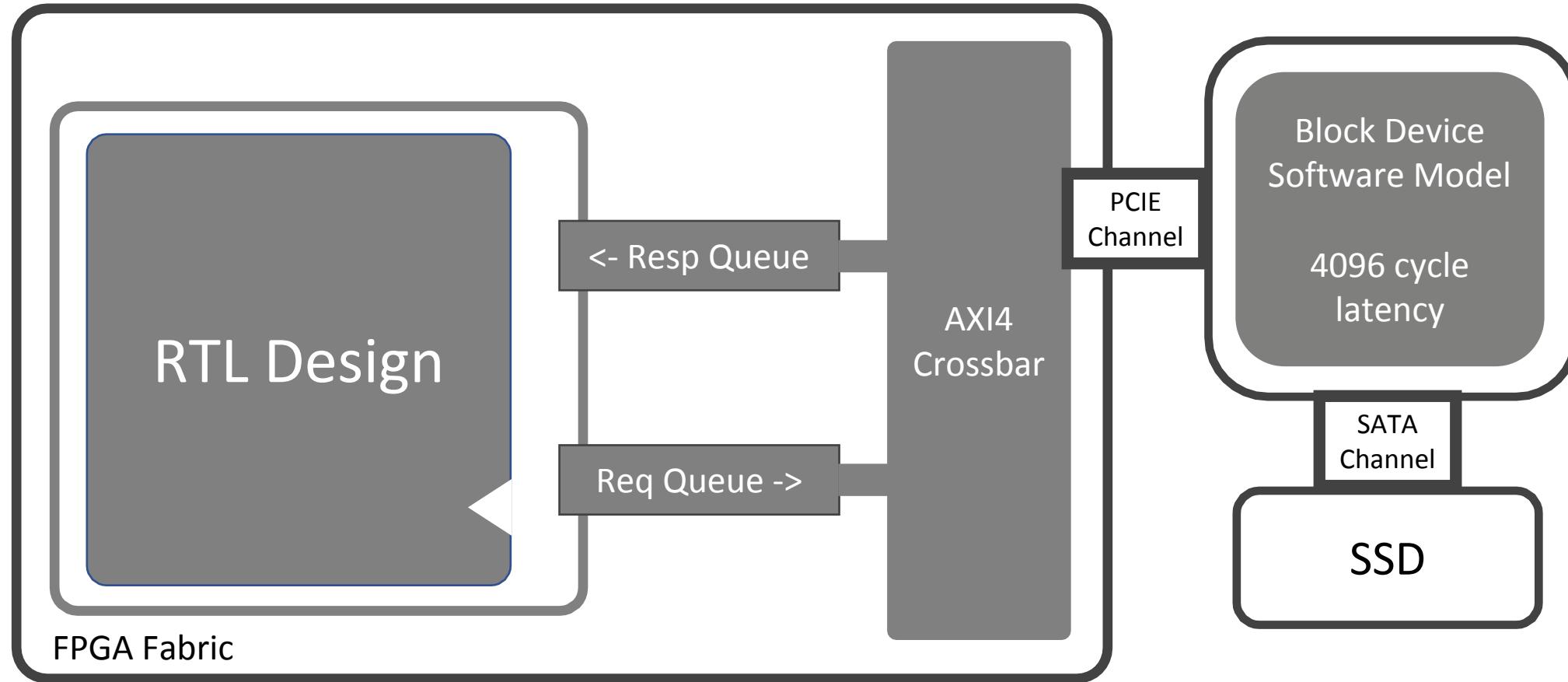


SoC sees realistic DRAM latency





# Host Decoupling in FireSim: HW/SW Co-Simulation



SoC sees deterministic SSD latency





# Benefits of Host Decoupling on FPGAs

## Simulations:

- Execute deterministically
- Produce identical results on different hosts (FPGAs & CPUs)

This enables support for:

1. SW co-simulation (e.g. block device, network models)
2. Simulating large targets over distributed hosts (FireSim, ISCA '18)
3. Non-invasive debugging and instrumentation (DESSERT, FPL '18)





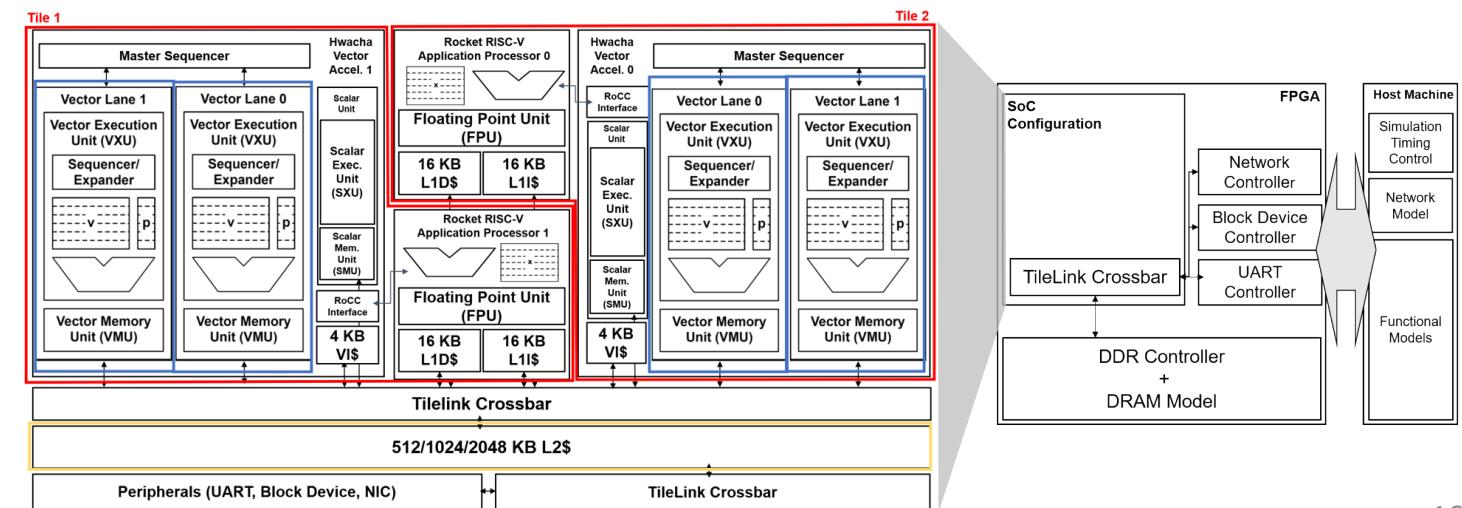
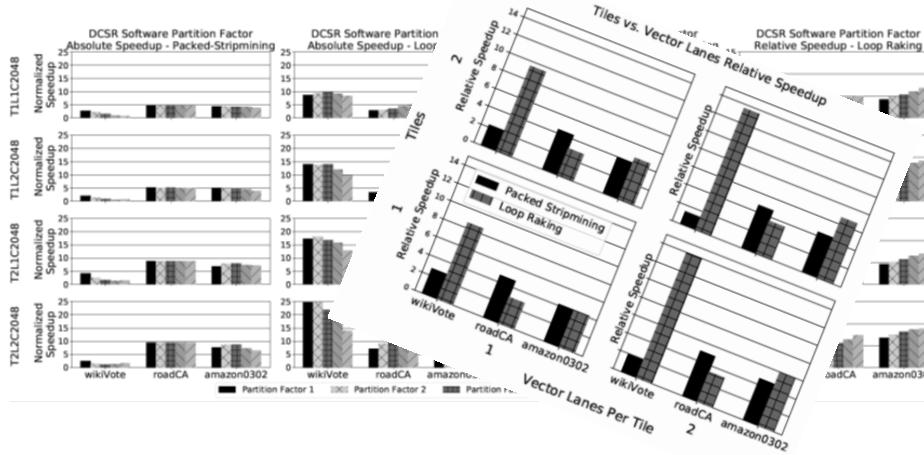
# What Can You Do With FireSim?





# Evaluating SoC Designs

- Performance
  - SPECInt2017 with reference inputs on Rocket Chip within a day
- Full-System Design Space Exploration
  - Data-parallel accelerators (Hwacha) and multi-core processors
  - Complex software stacks (Linux, OpenMP, GraphMat, Caffe)





# Evaluating SoC Designs

- Security
    - Replicate and identify uarch-level attacks, using pre-silicon RTL
    - BOOM Spectre replication
  - Accelerator prototyping (integrated with Rocket Chip)
    - Chisel-based ML accelerators
    - Open-source accelerator evaluation (NVDLA)
    - HLS-based rapid prototyping

# Replicating Spectre-v1/2

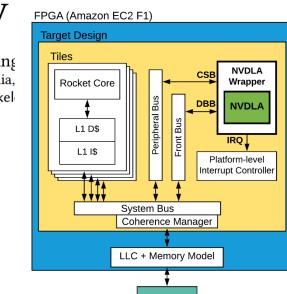
- Use
  - an
  - Rep

## BOOM Hardware Security Research

# Integrating NVIDIA Deep Learning Accelerator (NVDLA) with RISC-V

Farzad Farshchi  
University of Kansas  
farshchi@ku.edu

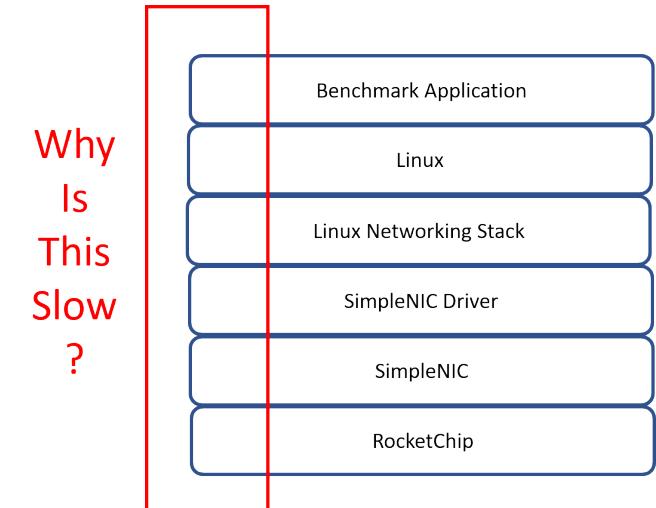
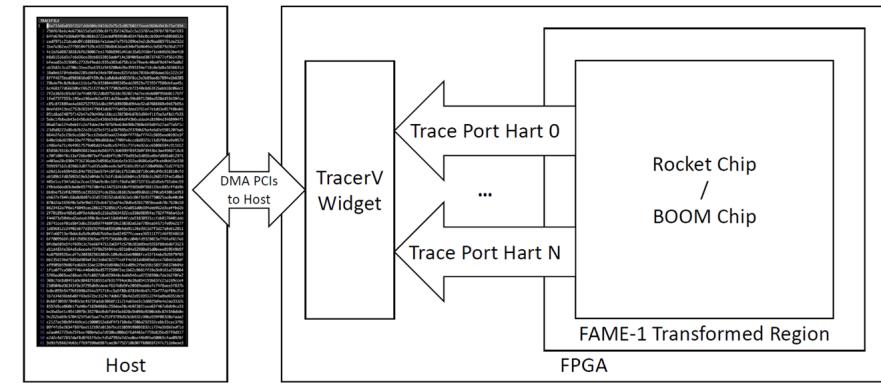
Qijing Huang  
University of California  
qijing.huang@berkeley.edu





# Debugging and Profiling SoC Designs

- TracerV: Widget for out-of-band collection of instruction traces from RISC-V systems
  - Profiling does not perturb execution
- Useful for kernel and hypervisor level cycle-sensitive profiling. Examples Use Cases:
  - Co-Optimization of NIC and Network Driver
  - Keystone Secure Enclave Project
  - High-performance hardware-specific code (supercomputing?)



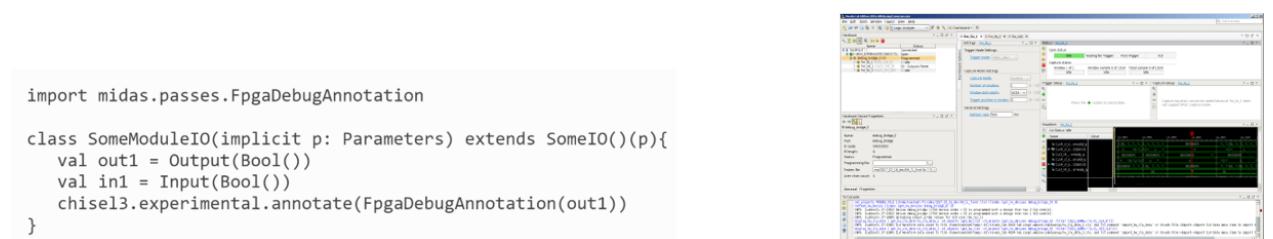


# Debugging and Profiling SoC Designs

- Autolla: Easy-to-use Integrated Logic Analyzer (ILA) support
  - User annotates interesting signals in the target design Chisel
  - Rest of the wiring is automatic
  - Use standard Vivado tools to control/collect data from ILA
- Example Autolla use cases:
  - Identify billion-cycle RVC instruction fetch bugs and load bugs in BOOM during Linux boot
  - Identifying AMO bug in the Hwacha accelerator deep into simulation (after Linux boot)

```
import midas.passes.FpgaDebugAnnotation

class SomeModuleIO(implicit p: Parameters) extends SomeIO()(p){
    val out1 = Output(Bool())
    val in1 = Input(Bool())
    chisel3.experimental.annotate(FpgaDebugAnnotation(out1))
}
```





# Debugging and Profiling SoC Designs

- Assertion Synthesis, Print Synthesis from DESSERT
  - Common software debugging primitives
  - Automatic integration with simulation host
  - Assertions helped identify BOOM bugs trillions of cycles into execution

**BROOM**  
An open-source out-of-order resilient low-voltage operation

Christopher Celio, Pi-Feng Chiu, Krste Asanović, David Patterson, and Borivoje Nikolic. Hot Chips 2018

RISC-V ASPIRE UC Berkeley

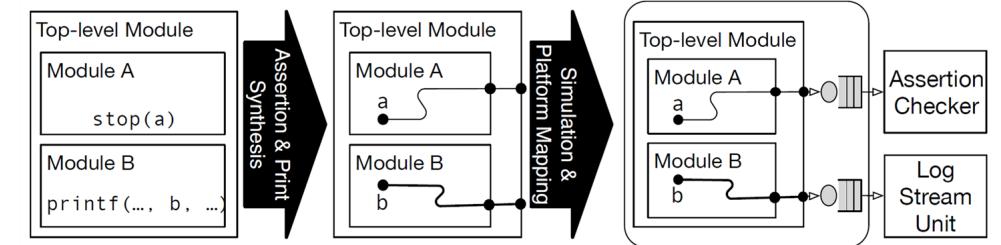
**BOOM-v2 Assertion Results**

Benchmark	Assertion	Cycle(B)	Simulation Time (Min)
483.xalancbmk.test	Invalid write back in ROB	1.9	3.4
464.h264ref.test	Pipeline hung	3.2	3.8
471.omnetpp.test	Pipeline hung	3.3	3.9
445.gobmk.test	Invalid write back in ROB	14.9	9.0
471.omnetpp.ref	Pipeline hung	62.6	22.2
401.bzip2.ref	Wrong JAL target	473.7	164.6

▪ Assertions are king.

▪ Cost: 2 x 50 cents / hour  
 ▪ Total cost: \$2 (compilation) + 2 x \$1.56 (simulation) = \$5.12

Benchmark	Assertion	Cycle(B)	Simulation Time (Min)
483.xalancbmk.test	Invalid write back in ROB	1.9	3.4
464.h264ref.test	Pipeline hung	3.2	3.8
471.omnetpp.test	Pipeline hung	3.3	3.9
445.gobmk.test	Invalid write back in ROB	14.9	9.0
471.omnetpp.ref	Pipeline hung	62.6	22.2
401.bzip2.ref	Wrong JAL target	473.7	164.6



DESSERT: Debugging RTL Effectively with State Snapshotting for Error Replays across Trillions of cycles

Donggyu Kim<sup>1</sup>, Christopher Celio<sup>2</sup>, Sagar Karandikar<sup>1</sup>, David Biancolin<sup>1</sup>, Jonathan Bachrach<sup>1</sup>, Krste Asanović<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering and Computer Sciences, University of California, Berkeley

{dgkim, sagark, biancolin, jrb, krste}@eecs.berkeley.edu

<sup>2</sup>Esperanto Technologies

christopher.celio@esperantotech.com

From: BROOM: An open-source Out-of-Order processor with resilient low-voltage operation in 28nm CMOS, Christopher Celio, Pi-Feng Chiu, Krste Asanovic, David Patterson and Borivoje Nikolic. HotChip 30, 2018

From: Donggyu Kim, Christopher Celio, Sagar Karandikar, David Biancolin, Jonathan Bachrach, and Krste Asanović, “DESSERT: Debugging RTL Effectively with State Snapshotting for Error Replays across Trillions of cycles”, FPL 2018





# How-to-build a *datacenter-scale* FireSim simulation

S. Karandikar et. al., "FireSim: FPGA-Accelerated Cycle-Exact Scale-Out System Simulation in the Public Cloud." *ISCA 2018*

S. Karandikar et. al., "FireSim: FPGA-Accelerated Cycle-Exact Scale-Out System Simulation in the Public Cloud." *IEEE Micro Top Picks 2018*





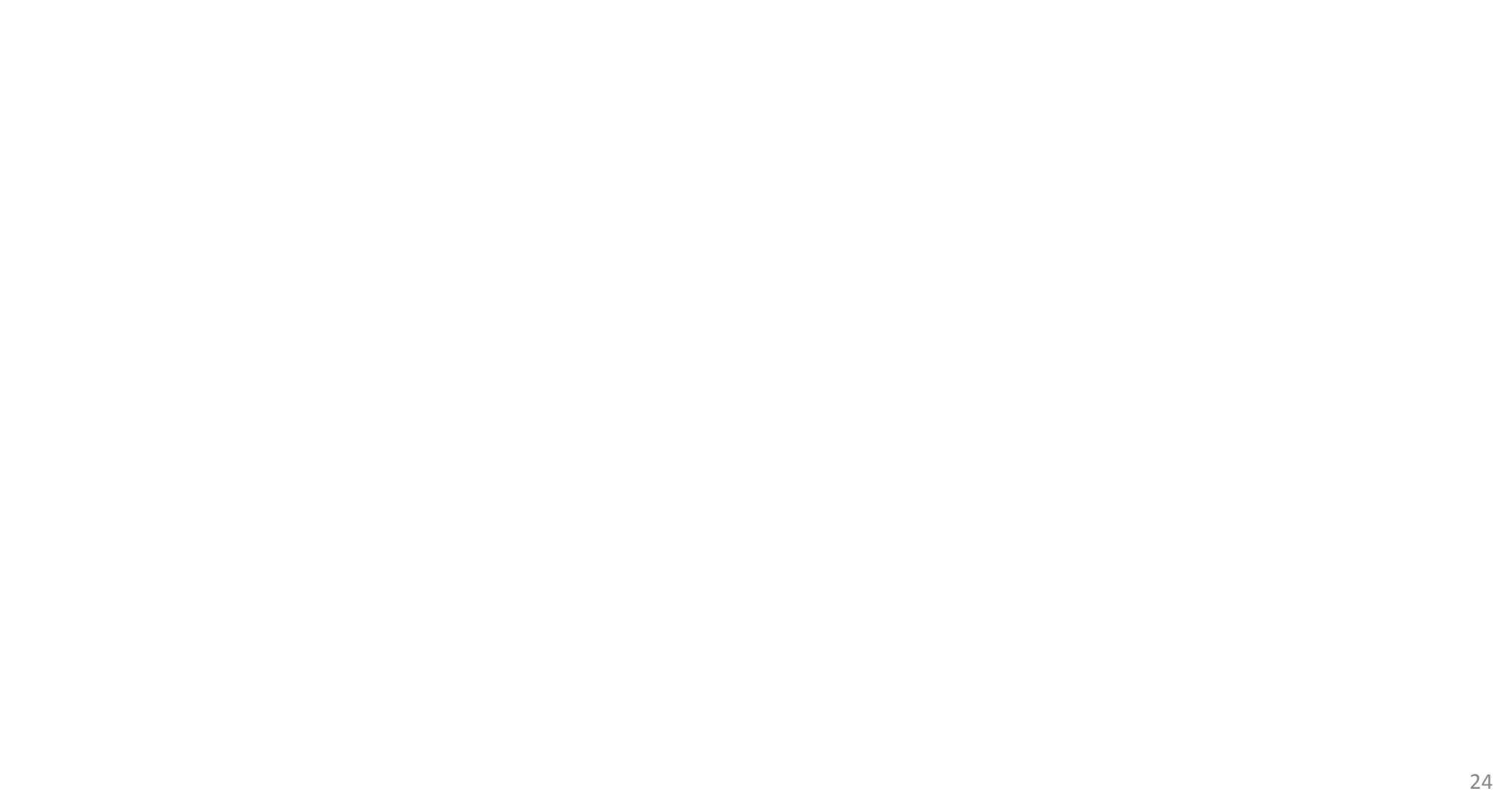
# Server SoC in RTL







# PGA Simulation of one server b



mulation of one server blade

Model  
- 4x R  
Cores  
- 16K  
- 256  
- 200  
NIC  
- 16 G  
Resou  
- < 1/4



# Step 3: FPGA Simulation of 4 server blades

## **Cost:**

\$0.49 per hour  
(spot)

\$1.65 per hour (on  
-demand)

## **Modeled System**

- 4 Server Blades
- 16 Cores
- 64 GB DDR3

## **Resource Util.**

- < 1 FPGA
- 4/4 Mem Chans

## **Sim Rate**

- ~14.3 MHz  
(netw)



# Step 3: FPGA Simulation of 4 server blades

## Modeled System

- 4 Server Blades
- 16 Cores
- 64 GB DDR3

## Resource Util.

- < 1 FPGA
- 4/4 Mem Chans

## Sim Rate

- ~14.3 MHz  
(netw)



# Step 4: Simulating a 32 node rack

## Cost:

\$2.60 per hour (spot)

\$13.20 per hour (on-demand)

## Modeled System

- 32 Server Blades
- 128 Cores
- 512 GB DDR3
- 32 Port ToR Switch
- 200 Gb/s, 2us links

## Resource Util.

- 8 FPGAs =
- 1x f1.16xlarge

## Sim Rate

- ~10.7 MHz (netw)



# Step 4: Simulating a 32 node rack

## Cost:

\$2.60 per hour (spot)

\$13.20 per hour (on-demand)

## Modeled System

- 32 Server Blades
- 128 Cores
- 512 GB DDR3
- 32 Port ToR Switch
- 200 Gb/s, 2us links

## Resource Util.

- 8 FPGAs =
- 1x f1.16xlarge

## Sim Rate

- ~10.7 MHz (netw)



# Step 4: Simulating a 32 node rack

## Modeled System

- 32 Server Blades
- 128 Cores
- 512 GB DDR3
- 32 Port ToR Switch
- 200 Gb/s, 2us links

## Resource Util.

- 8 FPGAs =
- 1x f1.16xlarge

## Sim Rate

- ~10.7 MHz (netw)



# Step 5: Simulating a 256 node “aggregation pod”

## Modeled System

- 256 Server Blades
- 1024 Cores
- 4 TB DDR3
- 8 ToRs, 1 Aggr
- 200 Gb/s, 2us links

## Resource Util.

- 64 FPGAs =
- 8x f1.16xlarge
- 1x m4.16xlarge

## Sim Rate

- ~9 MHz (netw)



# Step 5: Simulating a 256 node “aggregation pod”

## Modeled System

- 256 Server Blades
- 1024 Cores
- 4 TB DDR3
- 8 ToRs, 1 Aggr
- 200 Gb/s, 2us links

## Resource Util.

- 64 FPGAs =
- 8x f1.16xlarge
- 1x m4.16xlarge

## Sim Rate

- ~9 MHz (netw)



# Step 6: Simulating a 1024 node datacenter

## Modeled System

- 1024 Servers
- 4096 Cores
- 16 TB DDR3
- 32 ToRs, 4 Aggr, 1 Root
- 200 Gb/s, 2us links

## Resource Util.

- 256 FPGAs =
- 32x f1.16xlarge
- 5x m4.16xlarge

## Sim Rate

- ~6.6 MHz (netw)



# Step 6: Simulating a 1024 node datacenter

## Modeled System

- 1024 Servers

6 Cores

1TB DDR3

ToRs, 4 Aggr,  
10GbE ports

10Gb/s, 2us

source Util.

~250 FPGAs =

- 32x f1.16xlarge

- 5x m4.16xlarge

Sim Rate

- ~6.6 MHz (netw)

Harnesses ***millions of dollars*** of FPGAs  
to simulate ***1024 nodes cycle-exactly***  
with a cycle-accurate ***network simulation***  
and ***global synchronization***

at a cost-to-user of only ***100s of dollars/hour***



# And now, time for a demo





# Recap

- Host-decoupled FPGA-hosted models provide deterministic, cycle-accurate simulation at FPGA speeds
- FIRRTL transforms allow use of existing RTL without modification
- Co-simulated software models provide additional peripherals (block device, UART, network)
- Host-decoupling allows advanced debugging tools like print/assert synthesis and out-of-band profiling
- Network/switch model allows scale-out simulation of entire datacenter





# Questions?

(P.S. Come see me for free stickers!)



Berkeley Architecture Research

Learn More:

Web: <https://fires.im>

Docs: <https://docs.fires.im>

GitHub: <https://github.com/firesim/firesim>

 [@firesimproject](https://twitter.com/firesimproject)

ISCA'18 Paper (Overview and DC Sim):

[sagark.org/assets/pubs/firesim-isca2018.pdf](https://sagark.org/assets/pubs/firesim-isca2018.pdf)

Micro Top Picks 2018 (Summary + Updates):

<https://ieeexplore.ieee.org/document/8688441>

FPGA'19 Paper (Synth. DRAM Model):

[eecs.berkeley.edu/~biancolin/papers/fased-fpga19.pdf](https://eecs.berkeley.edu/~biancolin/papers/fased-fpga19.pdf)

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000849, and by DARPA, Award Number HR0011-12-2-0016. Research was also partially funded by ADEPT Lab industrial sponsors and affiliates Intel, Apple, Futurewei, Google, and Seagate, and RISE Lab sponsor Amazon Web Services. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.



# References

- [1] Peter X. Gao, Akshay Narayan, Sagar Karandikar, Joao Carreira, Sangjin Han, Rachit Agarwal, Sylvia Ratnasamy, and Scott Shenker. 2016. Network requirements for resource disaggregation. OSDI'16
- [2] Y. Lee *et al.*, "An Agile Approach to Building RISC-V Microprocessors," in *IEEE Micro*, vol. 36, no. 2, pp. 8-20, Mar.-Apr. 2016.
- [3] Jacob Leverich and Christos Kozyrakis. Reconciling high server utilization and sub-millisecond quality-of-service. EuroSys '14
- [4] Zhangxi Tan, Zhenghao Qian, Xi Chen, Krste Asanovic, and David Patterson. DIABLO: A Warehouse-Scale Computer Network Simulator using FPGAs. ASPLOS '15
- [5] Tan, Z., Waterman, A., Cook, H., Bird, S., Asanović, K., & Patterson, D. A case for FAME: FPGA architecture model execution. ISCA '10
- [6] Evaluation of RISC-V RTL Designs with FPGA Simulation. Donggyu Kim, Christopher Celio, David Biancolin, Jonathan Bachrach and Krste Asanovic. CARRV '17.
- [7] Donggyu Kim, Adam Izraelevitz, Christopher Celio, Hokeun Kim, Brian Zimmer, Yunsup Lee, Jonathan Bachrach, and Krste Asanović. Strober: fast and accurate sample-based energy simulation for arbitrary RTL. ISCA '16
- [8] Donggyu Kim, Christopher Celio, Sagar Karandikar, David Biancolin, Jonathan Bachrach, and Krste Asanović, "DESSERT: Debugging RTL Effectively with State Snapshotting for Error Replays across Trillions of cycles", FPL 2018

