FireSim and Chipyard Tutorial: Welcome!

You must enter a valid email on this form! →

1. Fill out the form at [in-person only] now for EC2 instance access

Then

2. You'll receive two emails. Follow insts to login, then wait.

Berkeley Architecture Research

FireSim

CHIPYARD

FireSim and Chipyard Tutorial: Intro

Sagar Karandikar

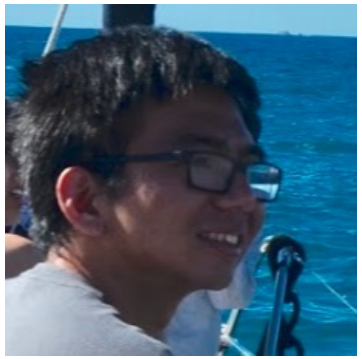
UC Berkeley

sagark@eecs.berkeley.edu

Berkeley Architecture Research

Presenters/Organizers

Sagar Karandikar



Jerry Zhao

Nayiri Krzysztofowicz

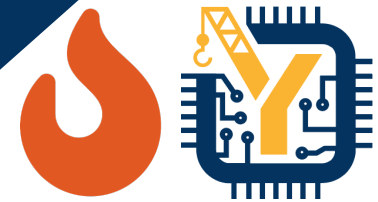
Abraham Gonzalez

Sophia Shao

Bora Nikolić

Krste Asanović

Getting Started/Logistics (recap)



- Fill out the form at **[in-person only]** now for EC2 instance access
- You'll receive two emails. One from Google Forms and one that looks like →
- Follow the instructions in this one to login to your FireSim manager instance, then wait

FireSim/Chipyard Tutorial User Info Inbox ×

FireSim Tutorial User Registration <mailgun@mg.sagark.org> 6:15 AM (1 minute ago) Reply
to sagark

Welcome to the FireSim/Chipyard tutorial!

Your Instance IP is 3.86.98.198
Your Instance Username is centos

There are two steps to login:

1) Save the attached key. You will likely need to fix permissions on it like so:

chmod 0600 tutorial-user-0000-us-east-1.pem

2) Next, there are two options for logging in, choose one. Mosh is highly recommended for easy persistent connections:

2a) If you have mosh installed (or can install it) we highly recommend logging in with mosh. See mosh install instructions here: https://mosh.org/#getting

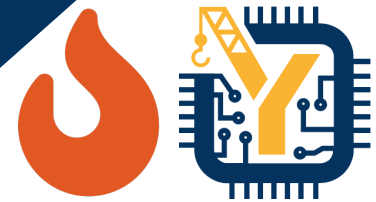
Once installed, to login with mosh, run:
mosh --ssh="ssh -i tutorial-user-0000-us-east-1.pem" centos@3.86.98.198

2b) If mosh is not available, login with a regular ssh client, then run screen once you're on the instance:
ssh -i tutorial-user-0000-us-east-1.pem centos@3.86.98.198
[now, start a screen on the remote instance]

Please let a presenter know if you have issues logging in.

tutorial-user-0000-...

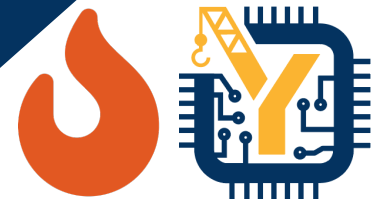
A *Golden Age* in Computer Architecture



- No more traditional scaling…
- An architect's dream: everyone wants custom microarchitectures and HW/SW co-designed systems
- Also, a golden age to have *direct impact* as researchers
  - Exploding open-source hardware environment
  - An open-ISA that can run software we care about

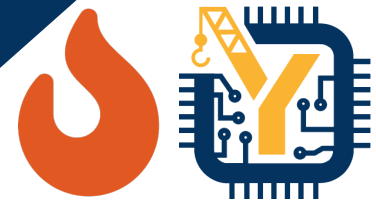
https://cacm.acm.org/magazines/2019/2/234352-a-new-golden-age-for-computer-architecture/fulltext

A *Dark Age* in Computer Architecture tools



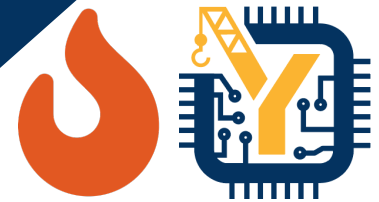
- What do we need to do good architecture research?
  - Need tools that let us evaluate designs on a variety of metrics:
    - Functionality
    - Performance
    - Power
    - Area
    - Frequency
  - Especially in small teams (grad students, startups), these tools need to be *agile*
  - Historically, without good open IP, had to build abstract arch/uarch simulators out of necessity
    - But now, we have much better IP and software compatibility, so what's stopping us?

A *Dark Age* in Computer Architecture tools



- Designed to be operated by hundreds of engineers
- Not, 10s of engineers or 1s-10s of grad students
- Two hard questions:
  - Where do I get a collection of well-tested hardware IP + complex software stacks that run on it?
  - How do I quickly obtain performance measurements for a novel HW/SW system?

Three hard questions, answered!



- Where do I get a collection of well-tested hardware IP + complex software stacks that run on it?

- How do I quickly obtain performance measurements for a novel HW/SW system?

- How do I get ASIC QoR feedback and tape out a design, with flexibility between open-source and proprietary flows

What can I do with these tools?

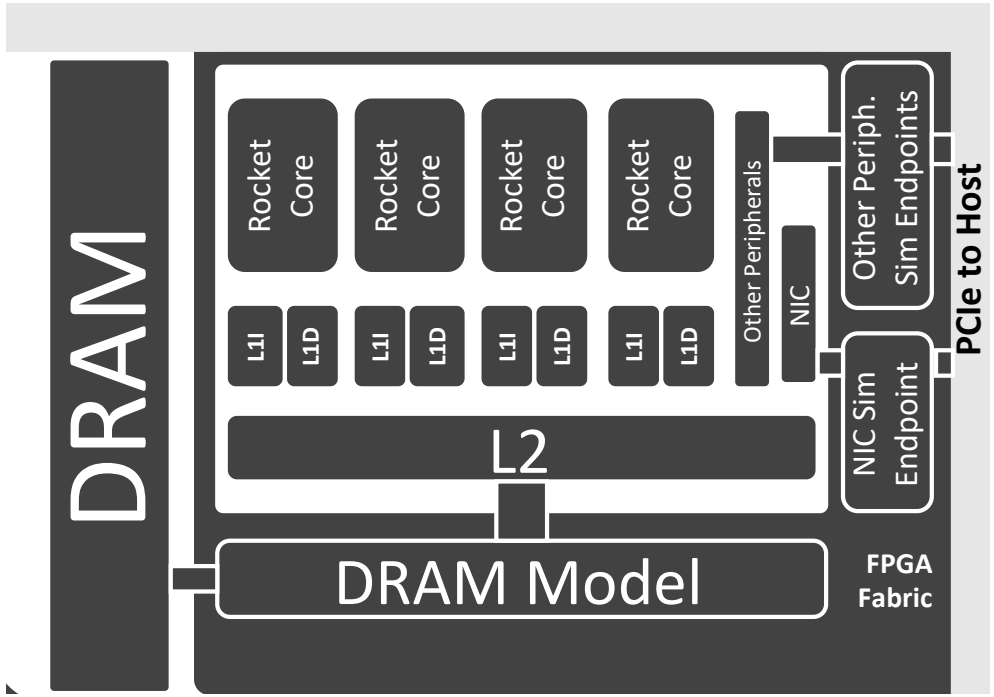
+

Hammer

Measure Functionality, Performance, Power, Area, Frequency *for real HW/SW systems*, quickly and easily, with small teams of engineers

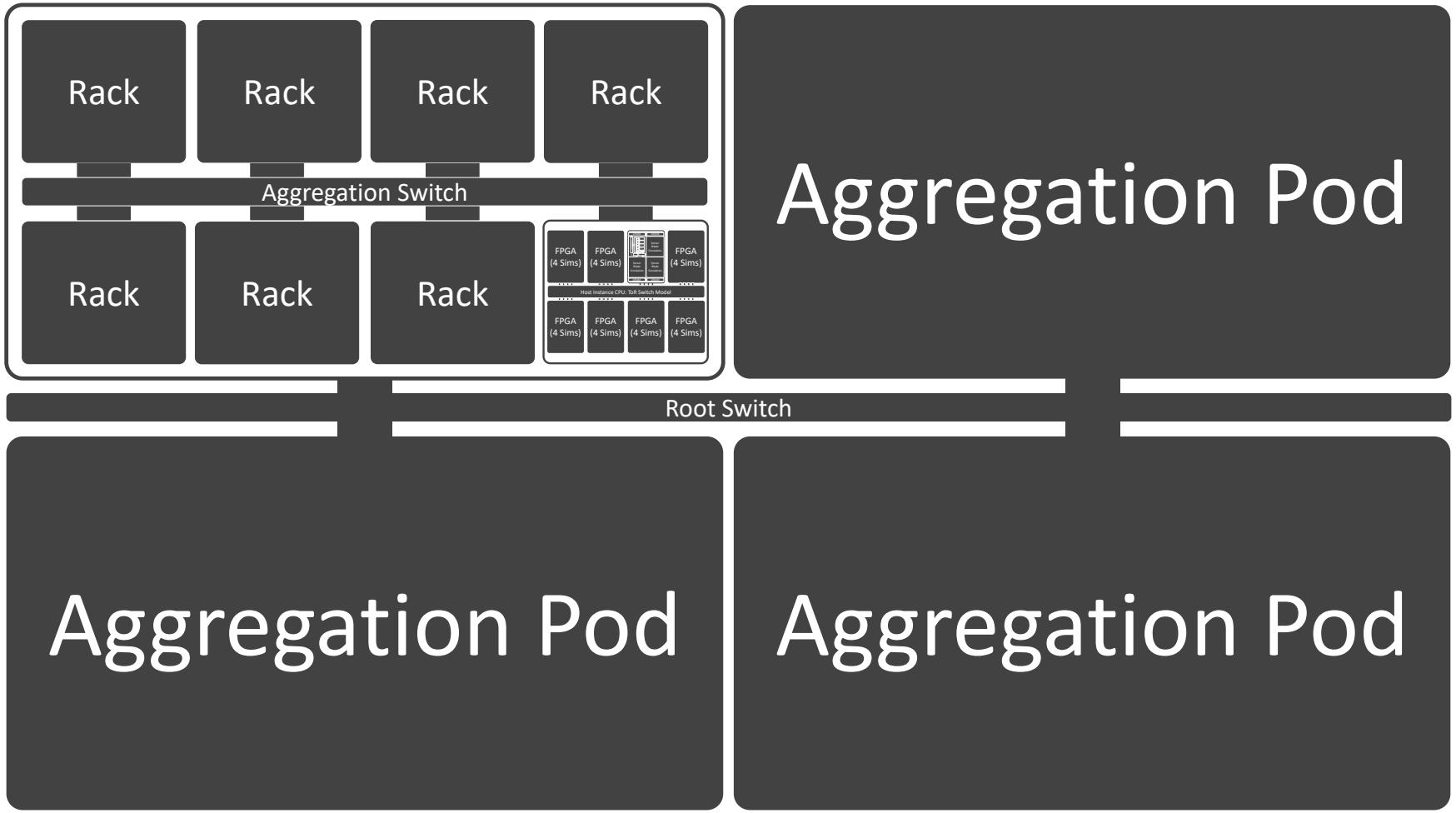
What kinds of designs can I work with?

- RISC-V Cores:
  - Rocket Chip In-Order core, industry proven
  - SonicBOOM Out-of-Order Superscalar core
  - CVA6
  - Ibex
- Accelerators
  - Hwacha Vector Accelerator
  - sha3 accelerator
  - NVDLA (NVIDIA Deep Learning Accelerator)
  - Gemmini (Berkeley DNN Accelerator)
  - FFT Generator
- Peripherals/other IP
  - L2 Cache, UART, Disk, Ethernet NIC, etc.
- FPGA-Simulation Models
  - Large LLCs, large DDR3 memory systems



Single SoC System

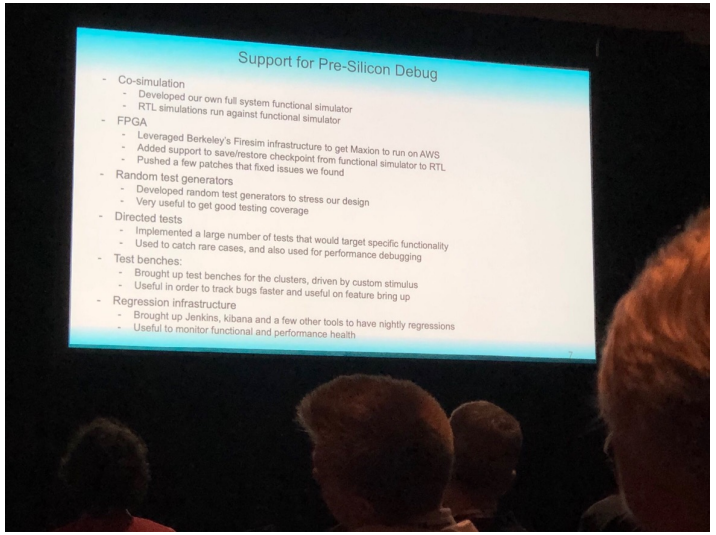
What kinds of designs can I work with?



Ethernet-Networked 1024 SoC System

Join the FireSim Community!: Open-source users and industrial users

- More than 190 mailing list members and 800 unique cloners per-week
- Projects with public FireSim support
  - Chipyard
  - Rocket Chip
  - BOOM
  - Hwacha Vector Accelerator
  - Keystone Secure Enclave
  - Gemmini
  - NVIDIA Deep Learning Accelerator (NVDLA):
    - NVIDIA blog post: https://devblogs.nvidia.com/nvdla/
  - BOOM Spectre replication/mitigation
  - Protobuf Accelerator
  - Too many to list here!
- Companies publicly announced using FireSim
  - Esperanto Maxion ET
  - Intensivate IntenCore
  - SiFive validation paper @ VLSI'20
  - Galois and Lockheed Martin (DARPA SSITH/FETT)



Esperanto announcement at RISC-V Summit 2018

FireSim in DARPA FETT

- DARPA SSITH: Building hardware defenses to address common software vulnerabilities
- DARPA FETT: How good are the defenses built in SSITH?
  - Multiple designs hosted for attack in FireSim [1]
- “Morpheus II: A RISC-V Security Extension for Protecting Vulnerable Software and Hardware”
  - Developed by UT Austin, U Mich., Agita Labs
  - Hosted on FireSim for FETT [2]
  - Over 500 attackers tried to break Morpheus II defenses, working for large bug bounties. None succeeded [3]

[1] K. Hopfer. Leveraging Amazon EC2 F1 Instances for Development and Red Teaming in DARPA’s First-Ever Bug Bounty Program. AWS APN Blog. May 2021.
[2] A. Harris, et. al., “Morpheus II: A RISC-V Security Extension for Protecting Vulnerable Software and Hardware”. In proceedings of the 2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), December 2021.
[3] T. Austin., et. al., “Morpheus II: A RISC-V Security Extension for Protecting Vulnerable Software and Hardware”. In HotChips 33, August 2021.

Join the FireSim Community!: Academic Users and Awards

- **ISCA ‘18**: Maas et. al. HW-GC Accelerator (**Berkeley**)
- **MICRO ‘18**: Zhang et. al. “Composable Building Blocks to Open up Processor Design” (**MIT**)
- **RTAS ‘20**: Farshchi et. al. BRU (**Kansas**)
- **EuroSys ‘20**: Lee et. al. Keystone (**Berkeley**)
- **OSDI ‘21**: Ibanez et. al. nanoPU (**Stanford**)
- **CCS ‘21**: Ding et. al. “Hardware Support to Improve Fuzzing Performance and Precision” (**Georgia Tech**)
- Too many to list here: see FireSim website for more!
  - https://fires.im/publications/#userpapers

- Awards: FireSim ISCA ‘18 paper:
  - IEEE Micro Top Pick
  - CACM Research Highlights Nominee from ISCA ’18
- Awards: FireSim users:
  - ISCA ‘18 Maas et. al.:
    - IEEE Micro Top Pick
  - MICRO ‘18 Zhang et. al.:
    - IEEE Micro Top Pick
  - MICRO ‘21 Gottschall et. al.:
    - MICRO-54 Best paper runner-up
  - MICRO ‘21 Karandikar et. al.:
    - MICRO-54 Distinguished Artifact winner
    - IEEE Micro Top Pick Honorable Mention
  - DAC ‘21 Genc et. al.:
    - DAC 2021 Best Paper winner

Join the FireSim Community!: Academic Users and Awards

- **ISCA '18**: Maas et. al. HW-GC Accelerator (**Berkeley**)
- **MICRO '18**: Zhang et. al. "Composable Building Blocks to Op
- **RTAS '20**: F
- **EuroSys '2**
- **OSDI '21**: Ib
- **CCS '21**: Di
  Improve Fuz
  (**Georgia Te**
- Too many to list here: see FireSim website for more!
  - https://fires.im/publications/#userpapers

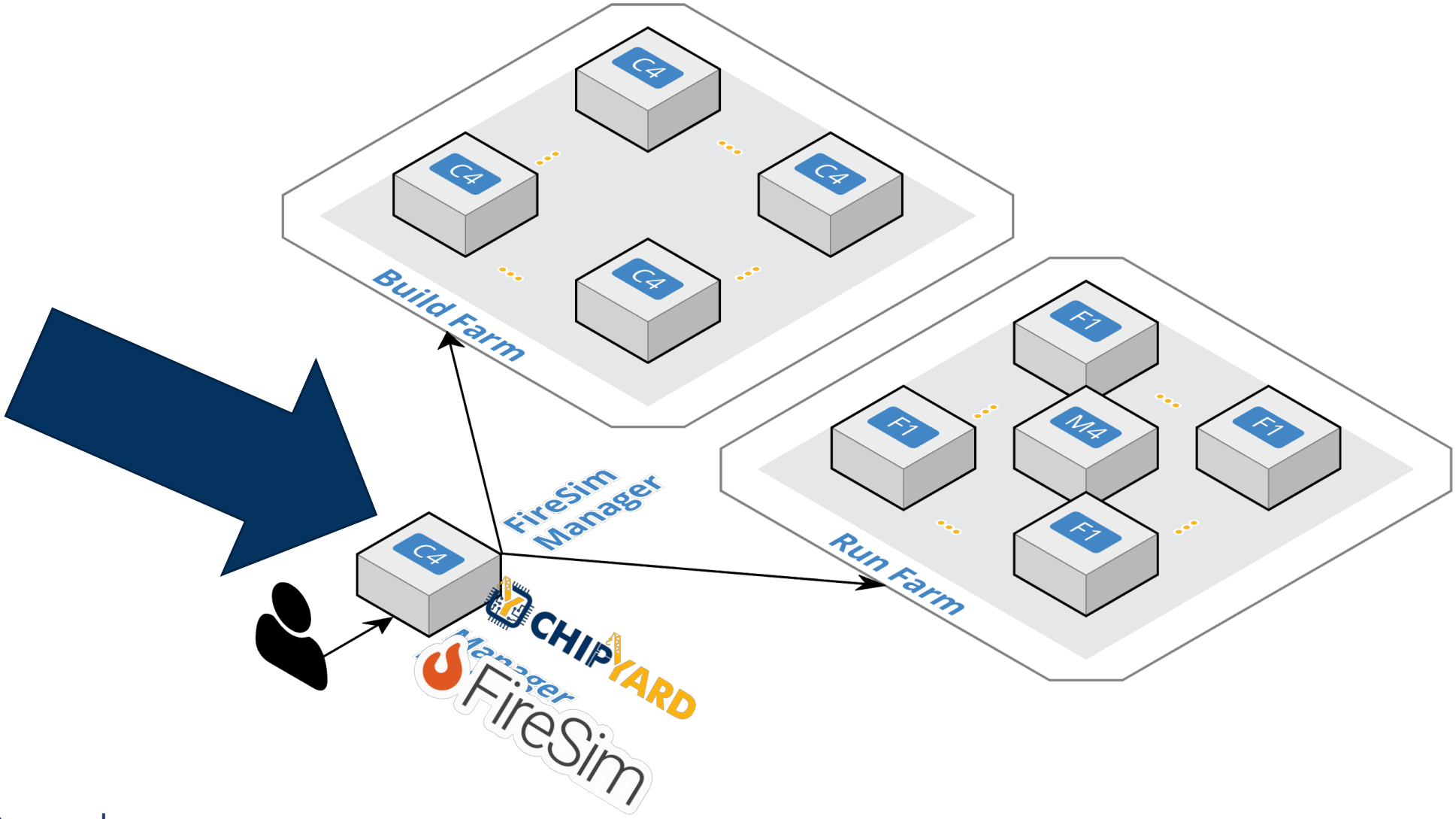
- Awards: FireSim ISCA '18 paper:
  - IEEE Micro Top Pick
  - CACM Research Highlights Nominee
  - winner
  - IEEE Micro Top Pick Honorable Mention
- DAC '21 Genc et. al.:
  - DAC 2021 Best Paper winner

FireSim has been used in published work from authors at over 20 academic and industrial institutions*

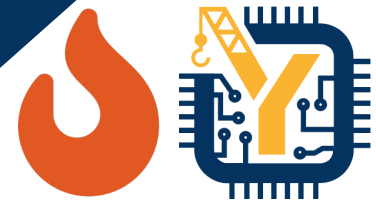
**actually used, not only cited*

Today's Logistics

You are here



Running a FireSim FPGA Build



- This will take a while, so we will run this in the background:

```
tmux new -s fpgabuild  # this will give you a persistent
                       # session you can reattach to
firesim managerinit --platform f1
```

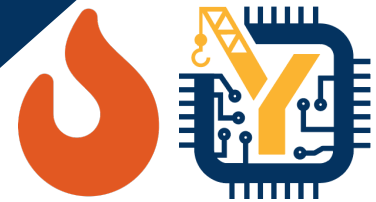
[When prompted, enter your email address to get a build completion notification]

```
# runs the HW build, all the way to AGFI
firesim buildbitstream
```

[Lastly, detach from tmux with “ctrl-b d”. We will return to this build later.]

[this will build a design called firesim_rocket_singlecore_no_nic_l2_lbp]

Today's Agenda - AM



8:00am: Introduction/Overview, Amazon EC2 Instance Setup, Logistics - Sagar

8:30am: Chipyard Basics – Jerry

9:00am: Building Custom RISC-V SoCs in Chipyard – Jerry

10:00am: Hammer VLSI flow - Nayiri

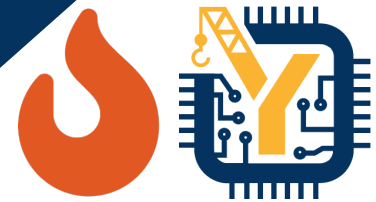
10:30am: Coffee break

11:00am: Running Hammer to Generate a GDS for a Chipyard SoC - Nayiri

11:30am: FPGA Prototyping - Nayiri

12:00pm - 1:30pm: Lunch

Today's Agenda - PM



1:30pm: FireSim Introduction - Sagar

2:00pm: Building Hardware Designs in FireSim – Sagar

2:30pm: Building Software Workloads in FireSim - Abe

3:00pm: Coffee break

3:30pm: Running a FireSim Simulation: Password Strength Checking on a RISC-V SoC with SHA-3 Accelerators and Linux - Sagar

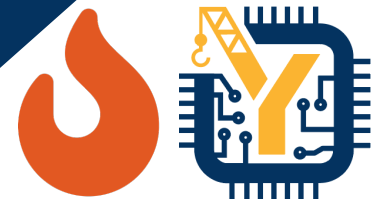
4:00pm: Debugging and Profiling FireSim-Simulated Designs - Abe

4:40pm: FireSim Local (On-Prem) FPGA and Distributed Metasimulation Support - Abe

4:55pm: Conclusion - Sagar

5:00pm: End of Tutorial

Thanks to AWS, Xilinx, and SLICE/ADEPT Lab Sponsors



SLICE Lab Sponsors:

National Science Foundation
WHERE DISCOVERIES BEGIN

NSF Award #2016662
CCRI: ENS: Chipyard