

Chipyard + FireSim

Generating and Evaluating Complex RISC-V SoCs

Abraham Gonzalez

EE290-2 Hardware for Machine Learning

Spring 2021

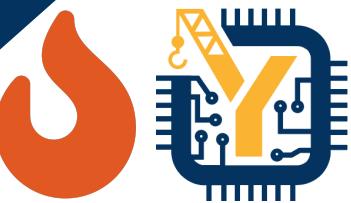
UC Berkeley



Berkeley
Architecture
Research

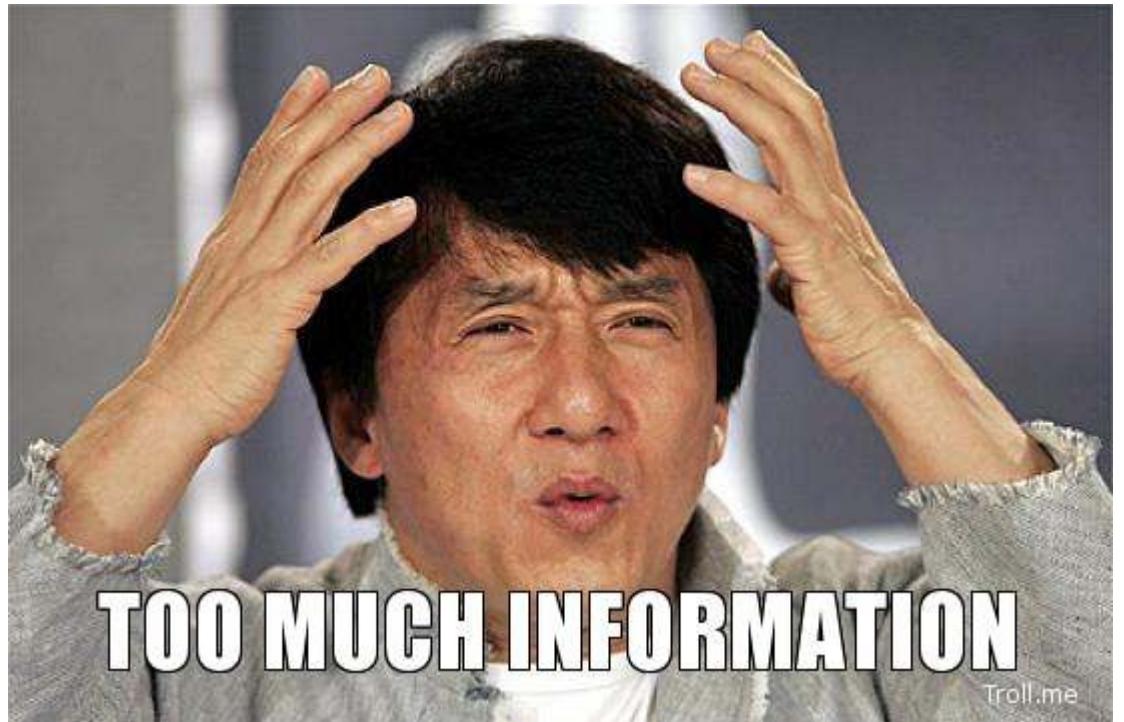
 FireSim

CHIPYARD

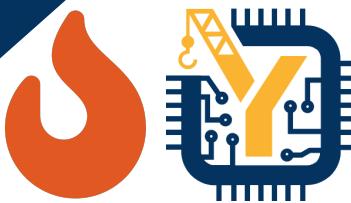


This Session

- Chipyard Introduction
- FireSim Introduction
- AWS Account Setup
- FireSim Initial Setup

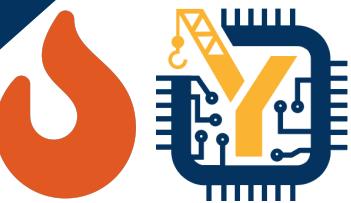


A **Golden Age** in Computer Architecture



- Traditional scaling is over
- Custom microarchitectures, HW/SW codesign enable performance gains through specialization
- Golden age for **open-source** across the entire stack
 - Open-source hardware
 - Open ISAs
 - Open-source operating systems, compilers, applications
- Let's build systems from open-source components, use custom IP for specialization



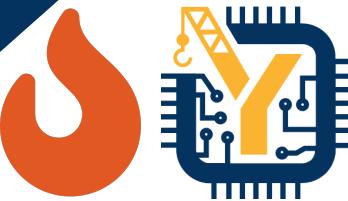


What are the Challenges?

- There's a lot of open IP ... how do I use any of it?
 - What do I do with a dump of open-source Verilog?
 - How to build a system out of mismatched components?
 - IP is only as good as the infrastructure around it
- How to obtain performance measurements quickly for my custom HW/SW system?
 - SW architectural simulators are slow and inaccurate
 - Hardware emulators are pricey, hard to use
 - Can't wait until tape-out to get design feedback
- How do I perform agile tape-outs with small teams?
 - Tape-out with 10s of engineers/graduate students



Two hard questions, answered!



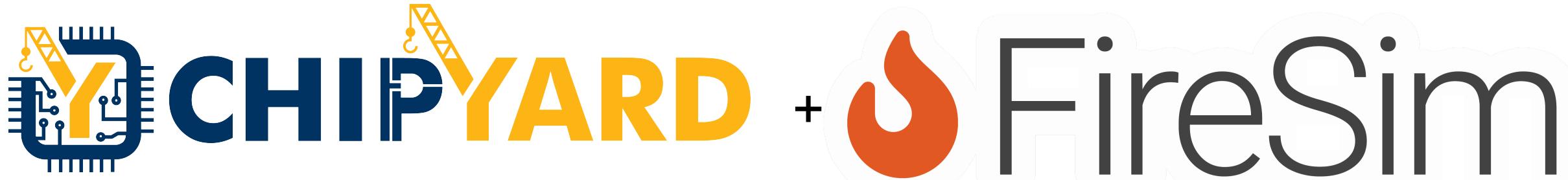
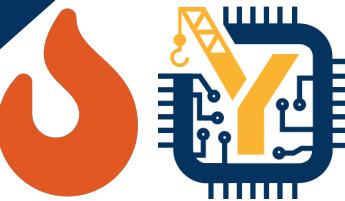
- Where do I get a collection of well-tested hardware IP + tooling + complex software stacks that run on it?



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



What can I do with these tools?



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



Berkeley Architecture Research

Chipyard Basics



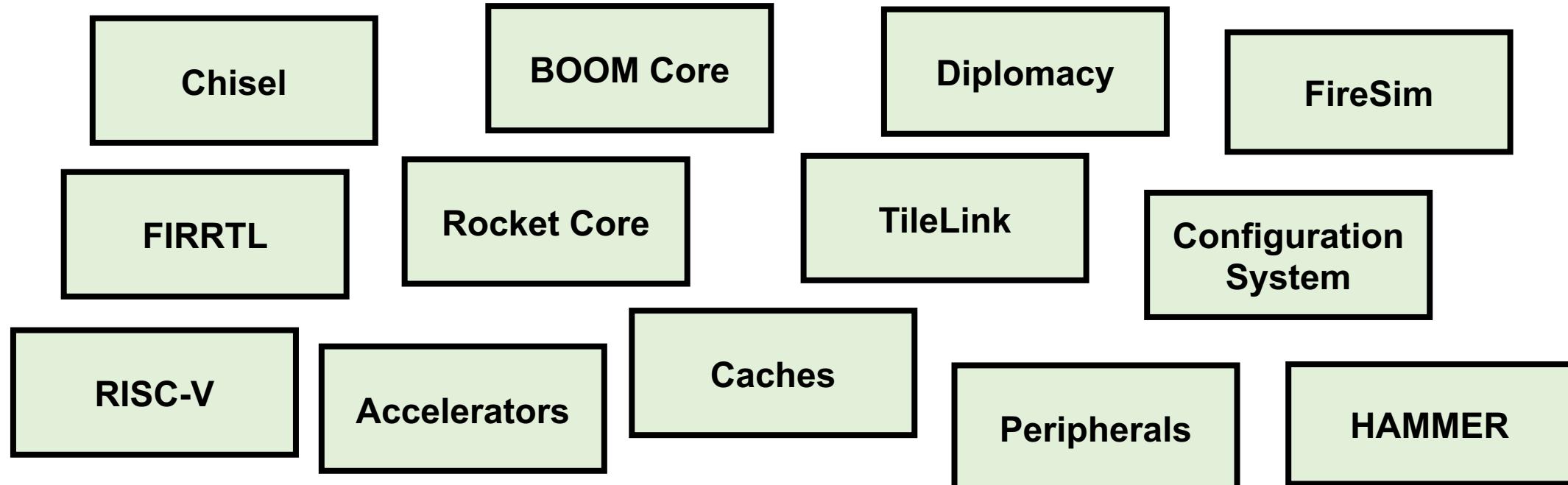
Berkeley
Architecture
Research

CHI PYARD

Berkeley Has Generated a Lot of Stuff!



Berkeley Architecture Research has developed and open-sourced:



But...

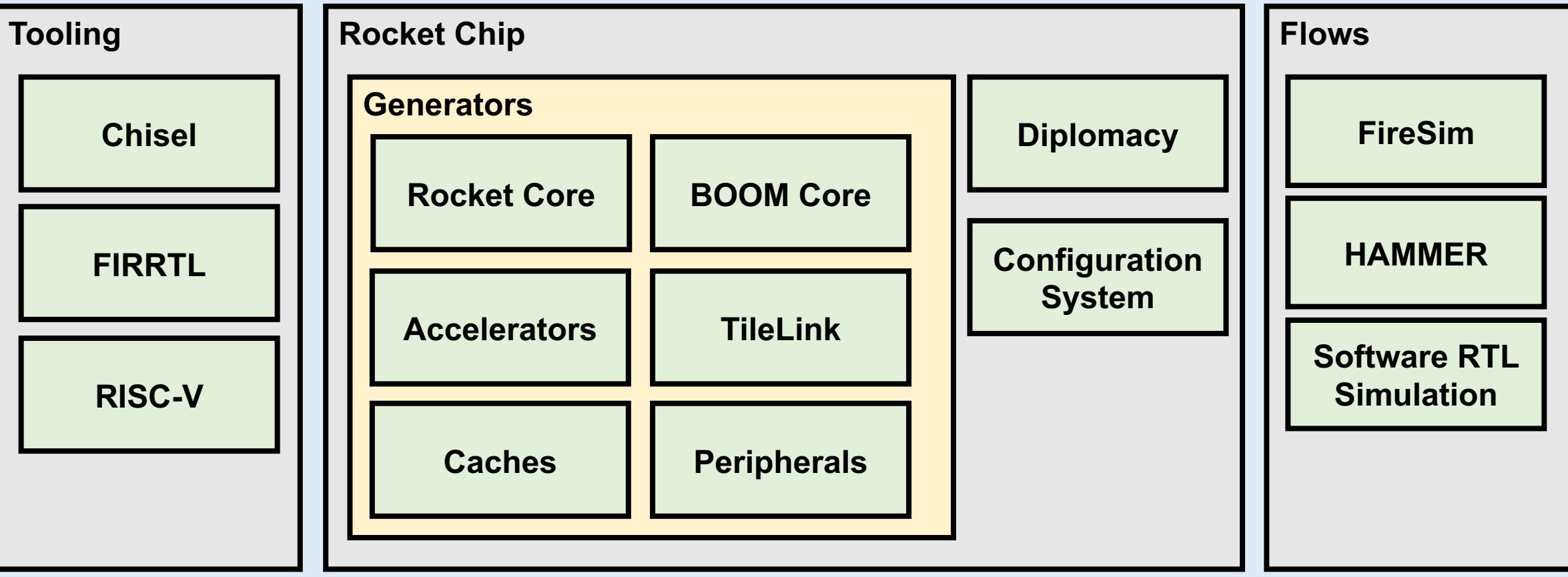
How do I put this all together to do something useful



Chipyard

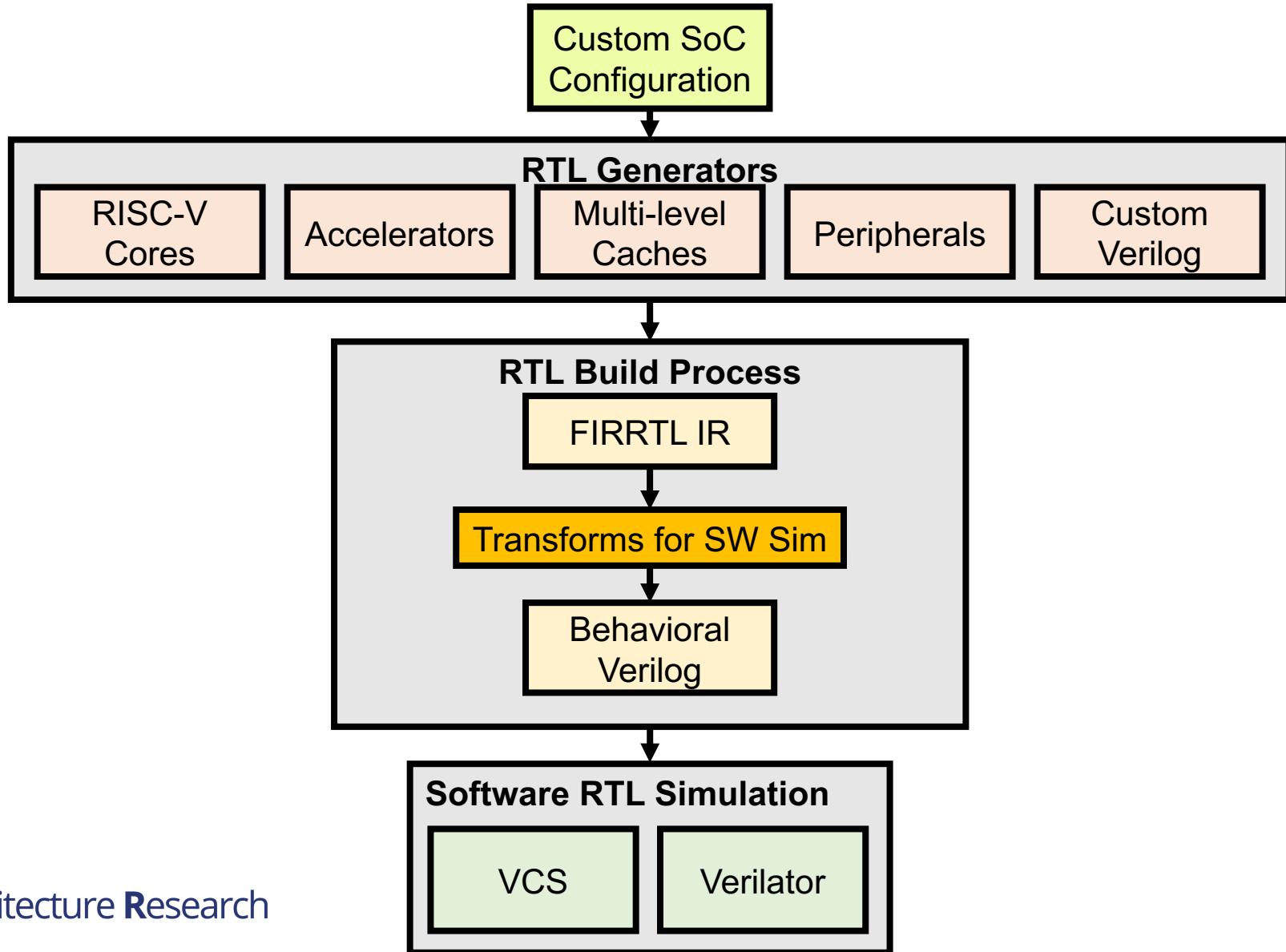


Chipyard



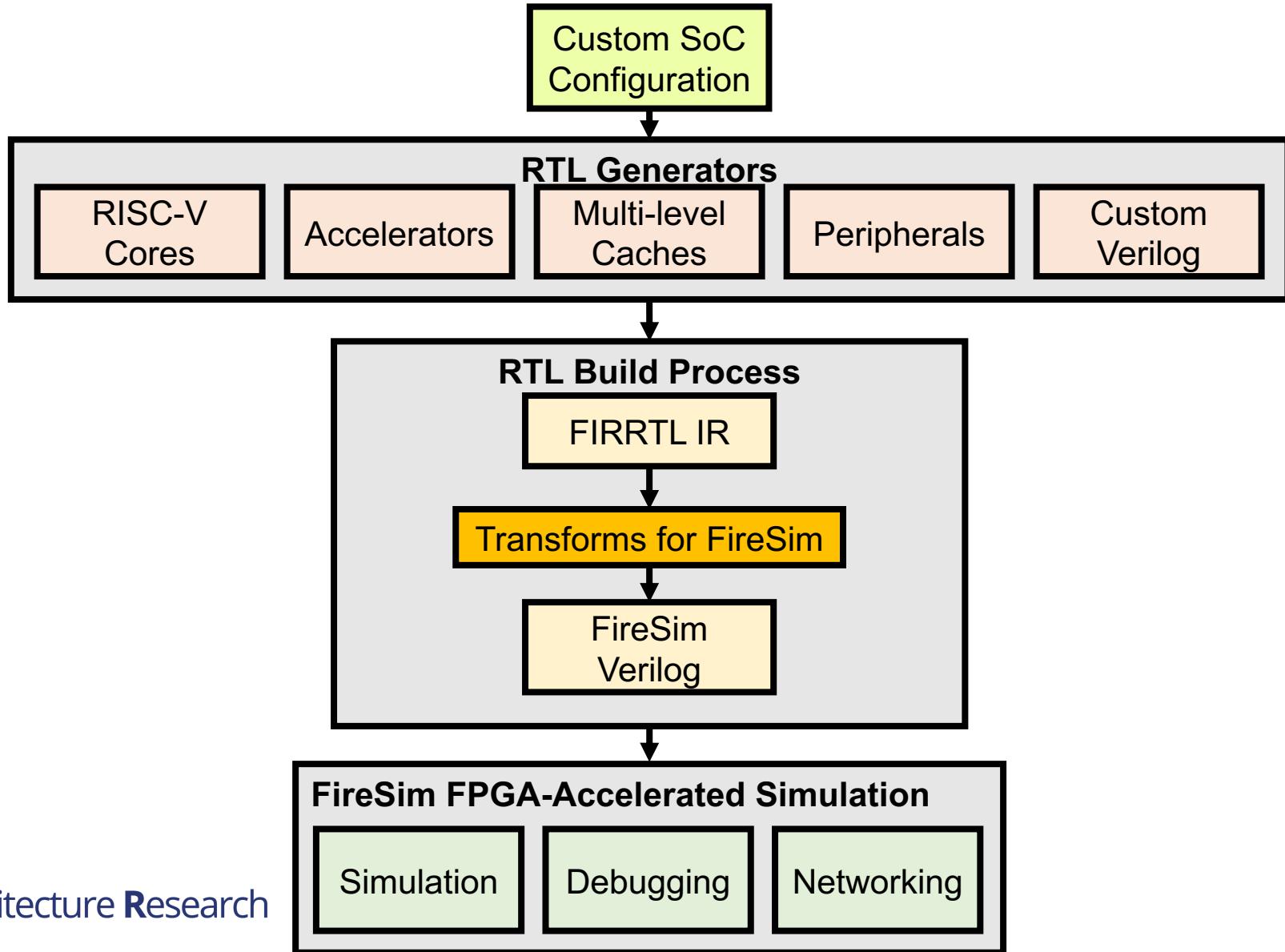


Chipyard SW RTL Simulation

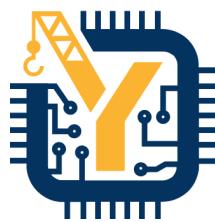




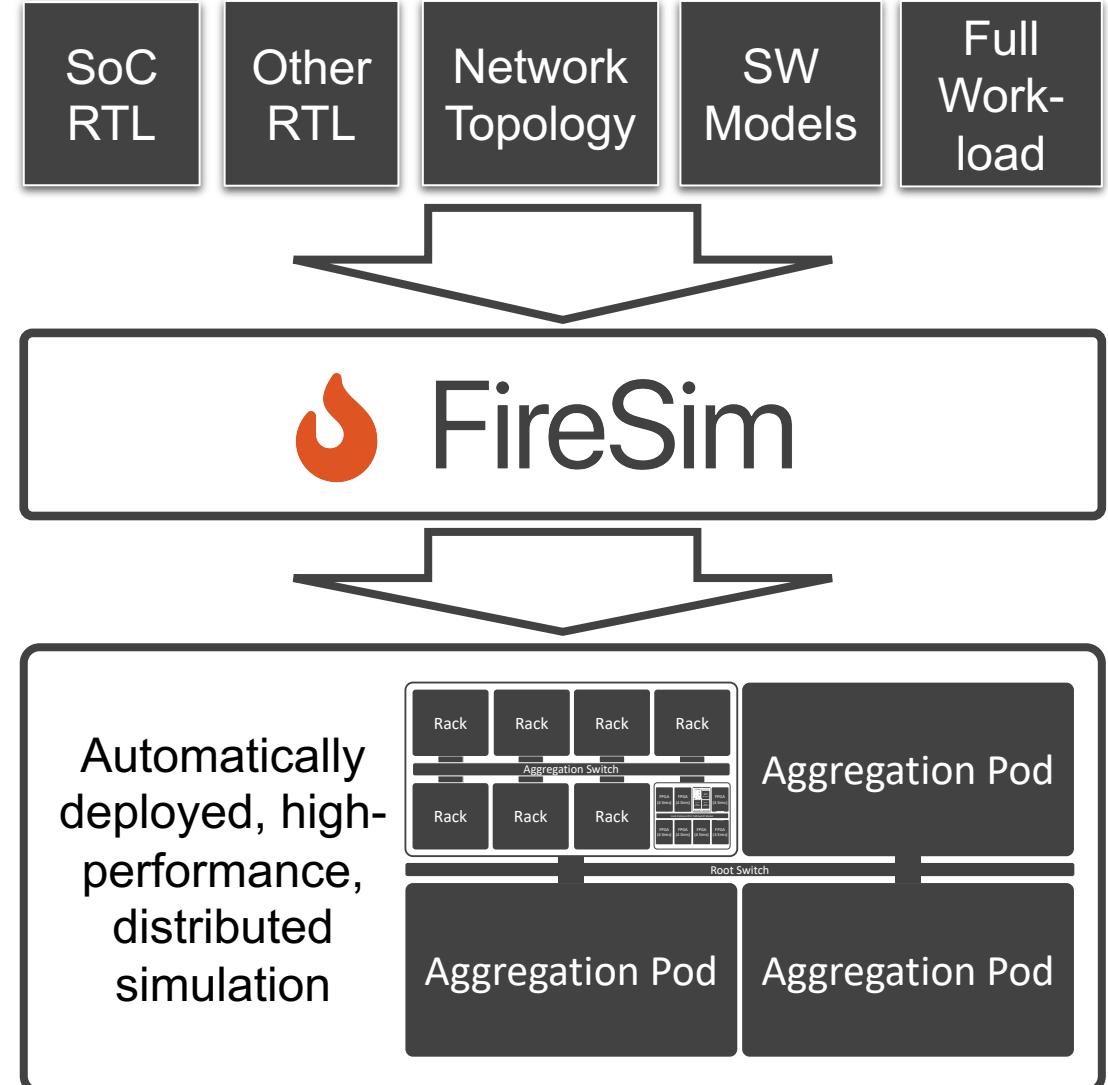
Chipyard targeting FireSim



FireSim

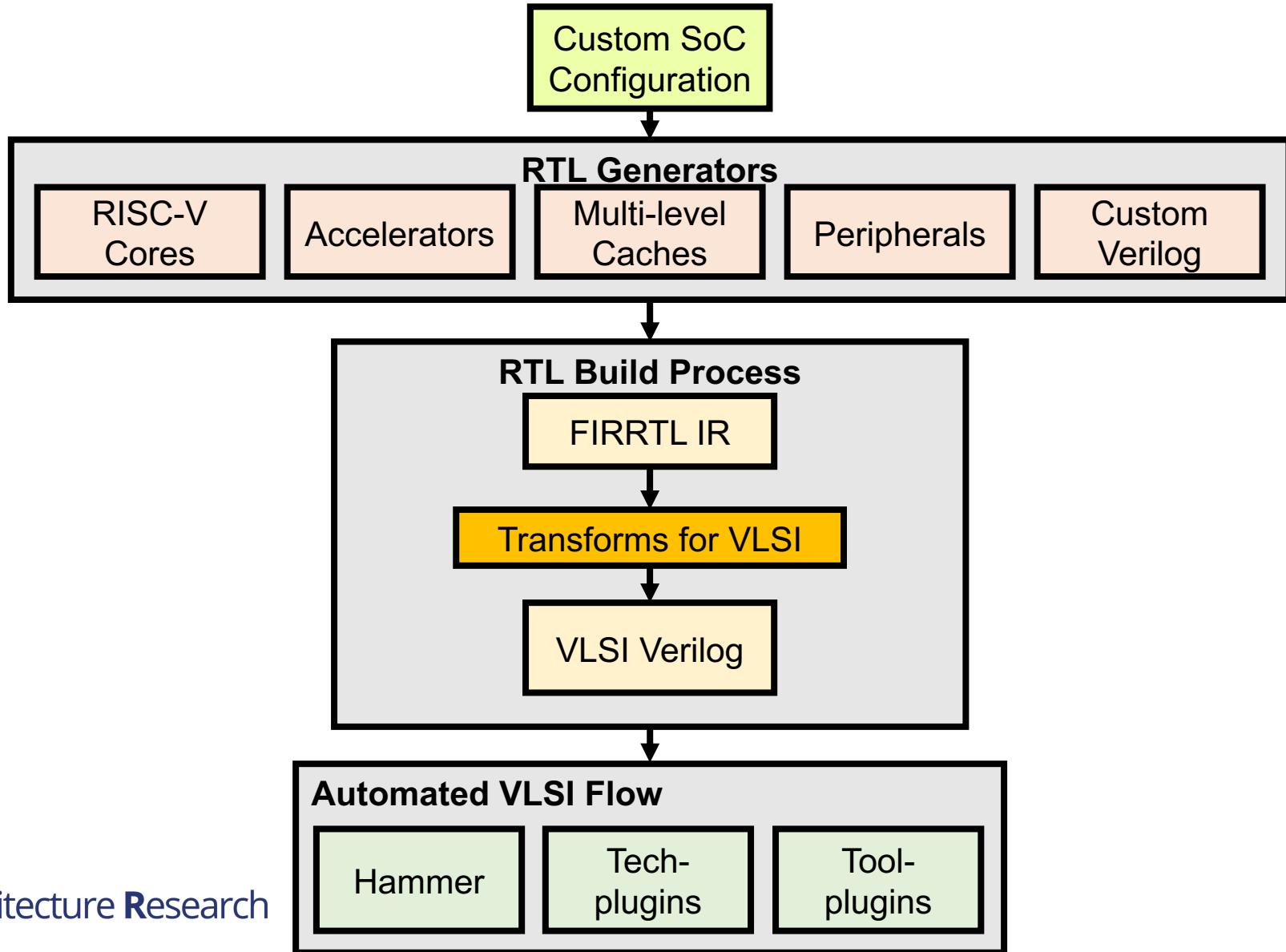


- FPGA-accelerated simulation on the Amazon EC2 public cloud
- Originally developed for cycle-exact architectural exploration of data-center clusters
- Not FPGA prototypes, rather FPGA-accelerated simulators
- Cloud FPGAs
 - Inexpensive, elastic supply of large FPGAs
 - Easy to collaborate with other researchers





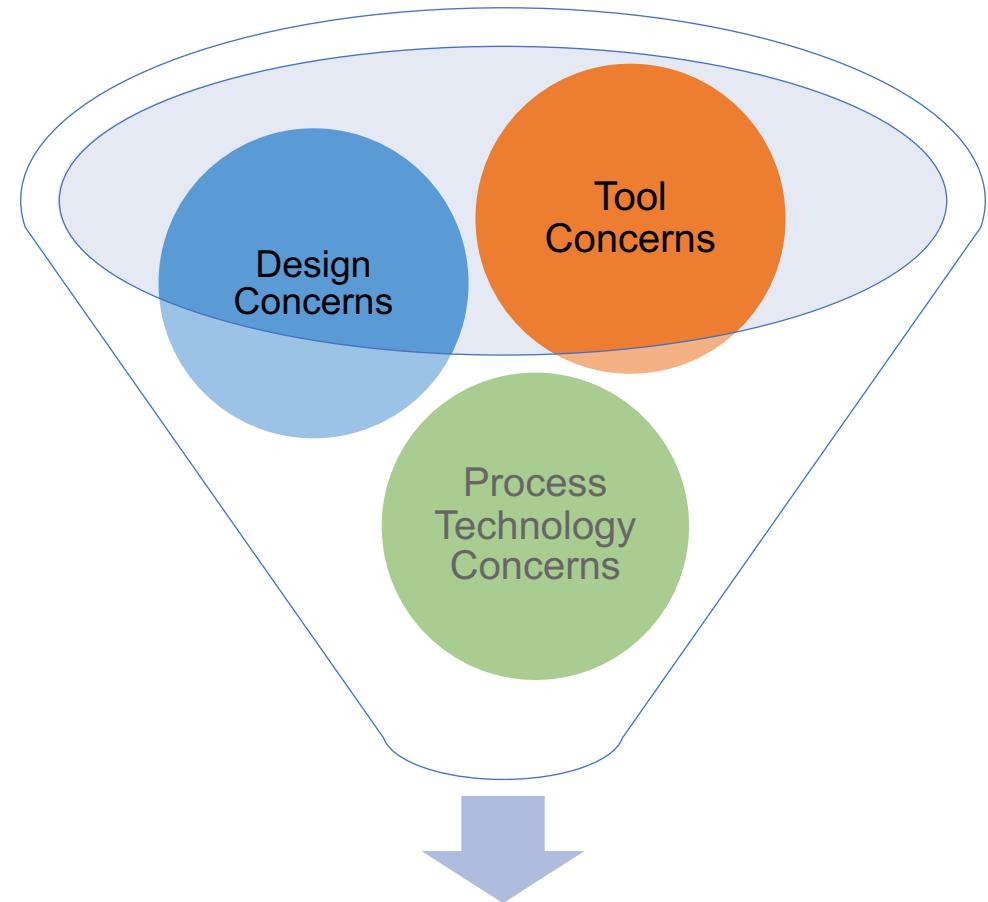
Chipyard VLSI Flow



Hammer



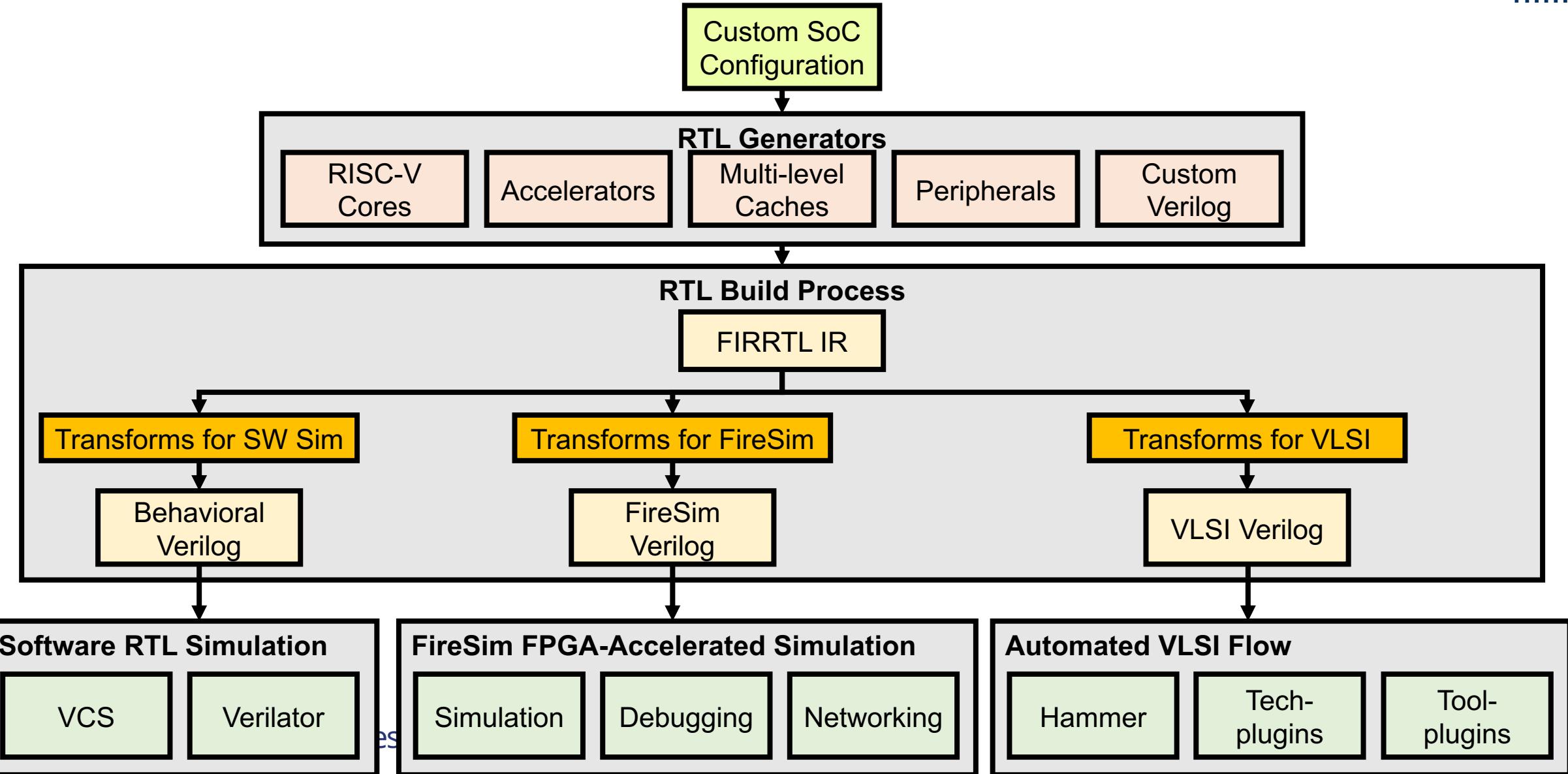
- Modular VLSI flow
 - Allow reusability
 - Allow for multiple “small” experts instead of a single “super” expert
 - Build abstractions/APIs on top
 - Improve portability
 - Improve hierarchical partitioning
- Three categories of flow input
 - Design-specific
 - Tool/Vendor-specific
 - Technology-specific



Magic TCL Script



Chipyard Unified Flows



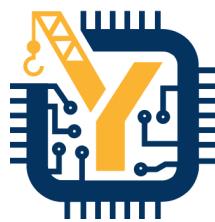
Chipyard Tooling



Berkeley
Architecture
Research

CHI PYARD

Chisel



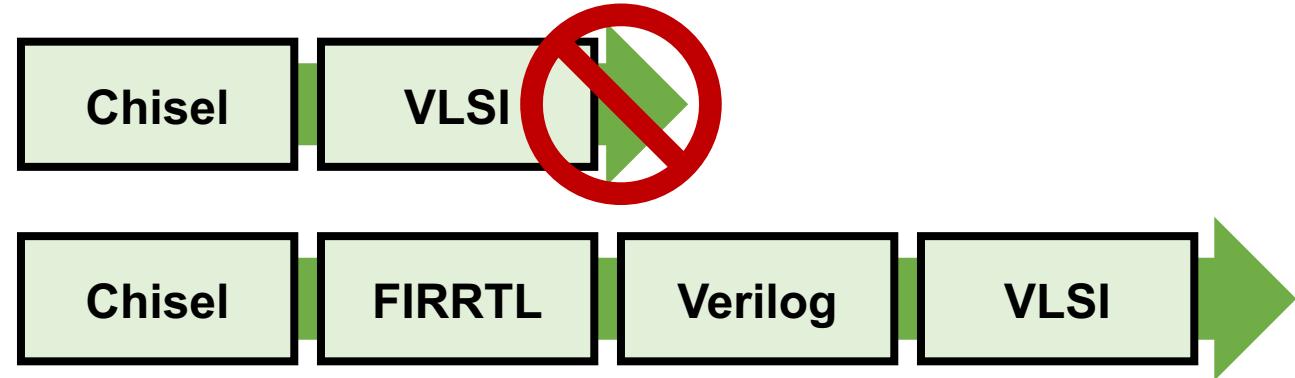
- Chisel – Hardware Construction Language built on Scala

- What Chisel **IS NOT**:

- **NOT** Scala-to-gates
- **NOT** HLS
- **NOT** tool-oriented language

- What Chisel **IS**:

- Productive language for **generating** hardware
- Leverage **OOP/Functional programming** paradigms
- Enables design of **parameterized generators**
- **Designer-friendly**: low barrier-to-entry, high reward
- **Backwards-compatible**: integrates with Verilog black-boxes



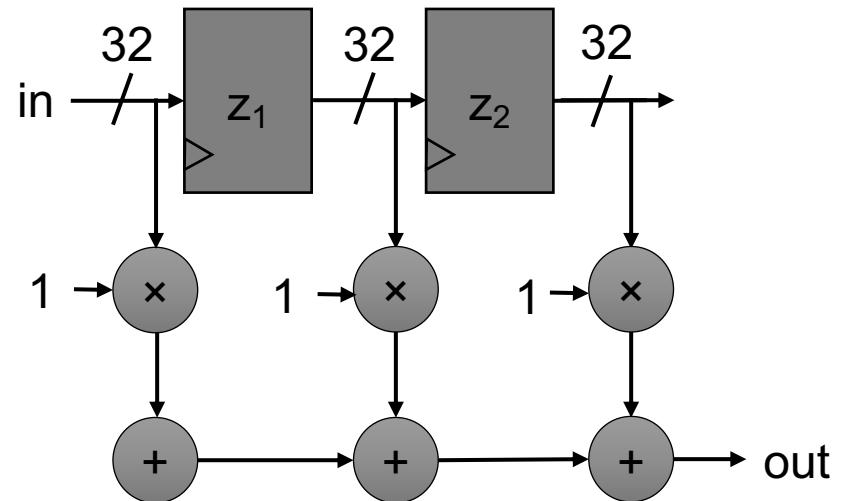


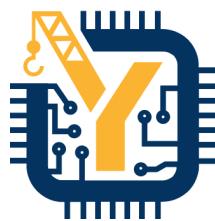
Chisel Example

```
// 3-point moving average implemented in the
// style of a FIR filter

class MovingAverage3 extends Module {
    val io = IO(new Bundle {
        val in = Input(UInt(32.W))
        val out = Output(UInt(32.W))
    })
    val z1 = RegNext(io.in)
    val z2 = RegNext(z1)

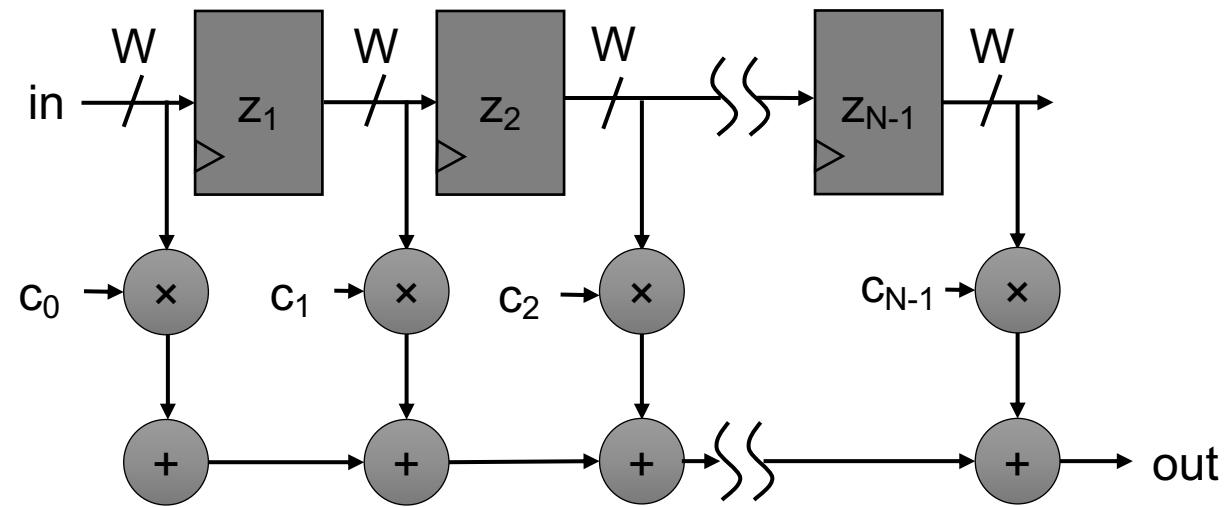
    io.out := io.in + z1 + z2
}
```



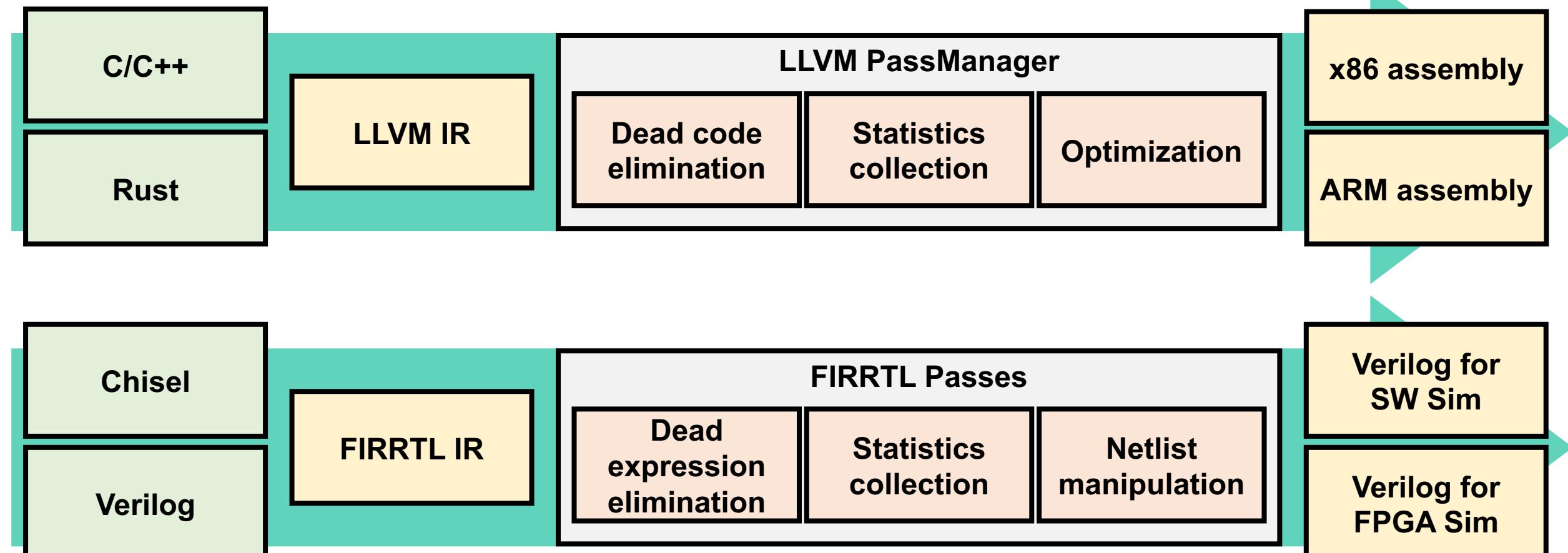


Chisel Example

```
// Generalized FIR filter parameterized by coefficients
class FirFilter(bitWidth: Int, coeffs: Seq[Int]) extends Module {
    val io = IO(new Bundle {
        val in = Input(UInt(bitWidth.W))
        val out = Output(UInt(bitWidth.W))
    })
    val zs = Wire(Vec(coeffs.length, UInt(bitWidth.W)))
    zs(0) := io.in
    for (i <- 1 until coeffs.length) {
        zs(i) := RegNext(zs(i-1))
    }
    val products = zs zip coeffs map {
        case (z, c) => z * c.U
    }
    io.out := products.reduce(_ + _)
}
```



FIRRTL - LLVM for Hardware



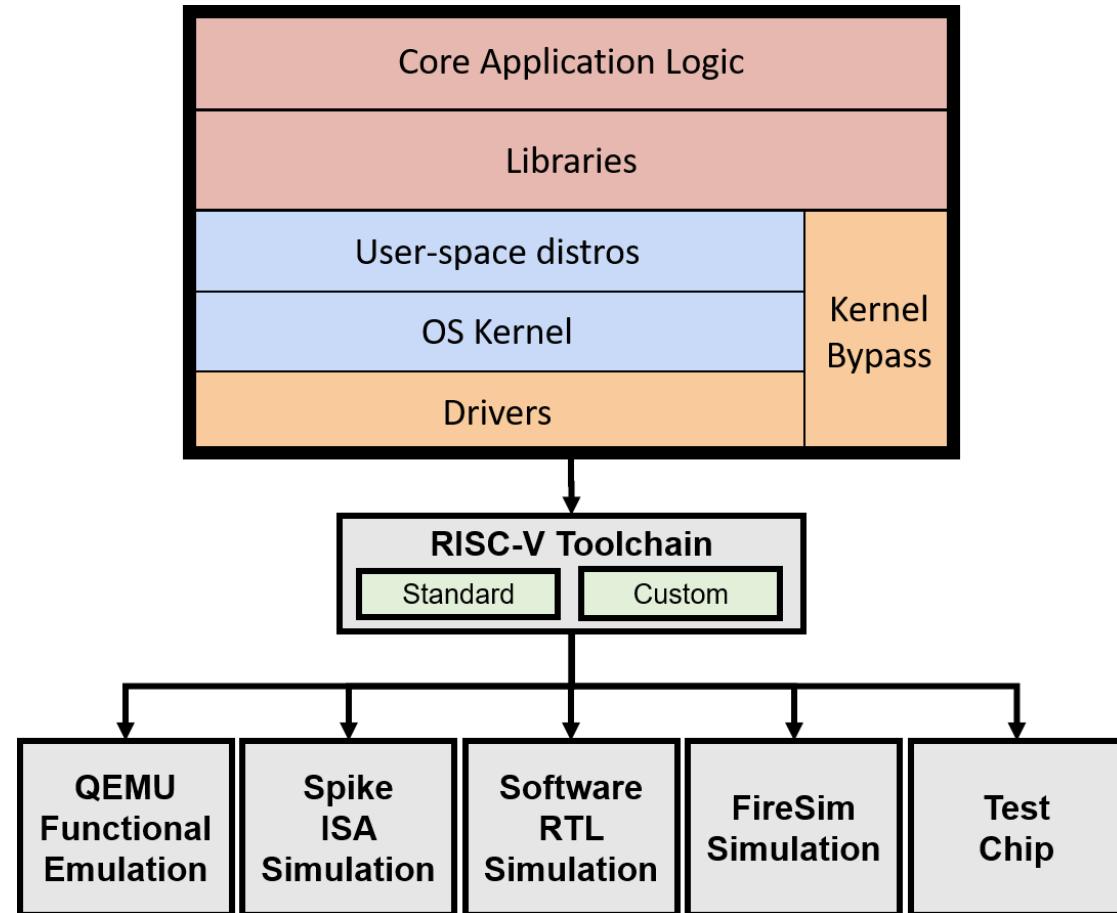
FIRRTL emits **tool-friendly, synthesizable** Verilog





Software

- Compatible standard RISC-V Tools versions
- ESP-Tools as a non-standard equivalent SW tools package with custom accelerator extensions (Hwacha, Gemmini)
- Improved BareMetal testing flow
 - Use libgloss and newlib instead of in-house syscalls
- FireMarshal workload management



Rocket Chip Generators



Berkeley
Architecture
Research

CHIPYARD



What is Rocket Chip?

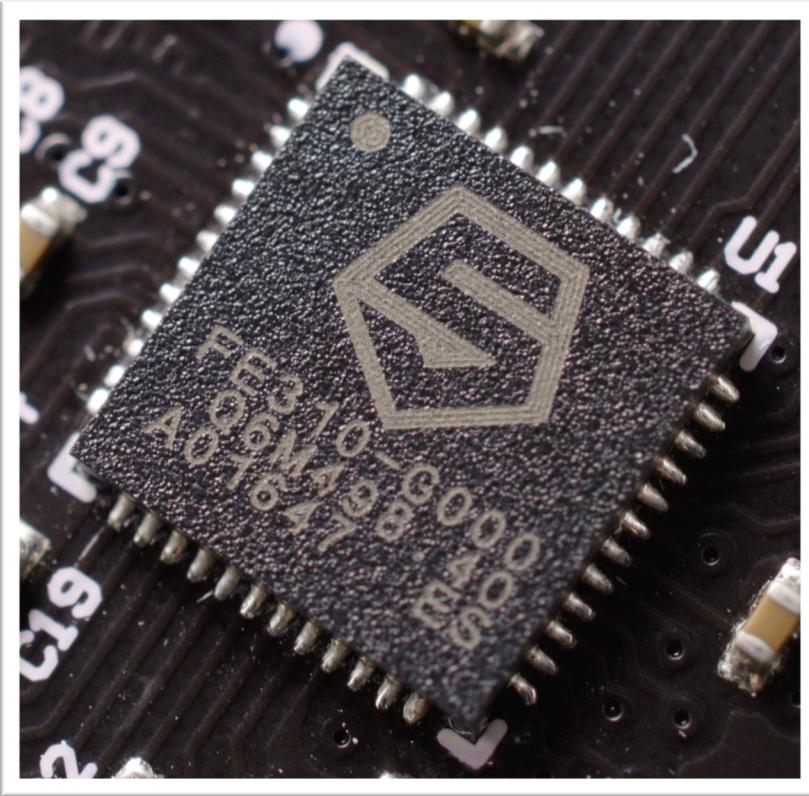
- A highly parameterizable and modular SoC generator
 - Replace default Rocket core w/ your own core
 - Add your own coprocessor
 - Add your own SoC IP to uncore
- A library of reusable SoC components
 - Memory protocol converters
 - Arbiters and Crossbar generators
 - Clock-crossings and asynchronous queues
- The largest open-source Chisel codebase
- Developed at UC Berkeley, now maintained by many
 - SiFive, CHIPS Alliance, UC Berkeley



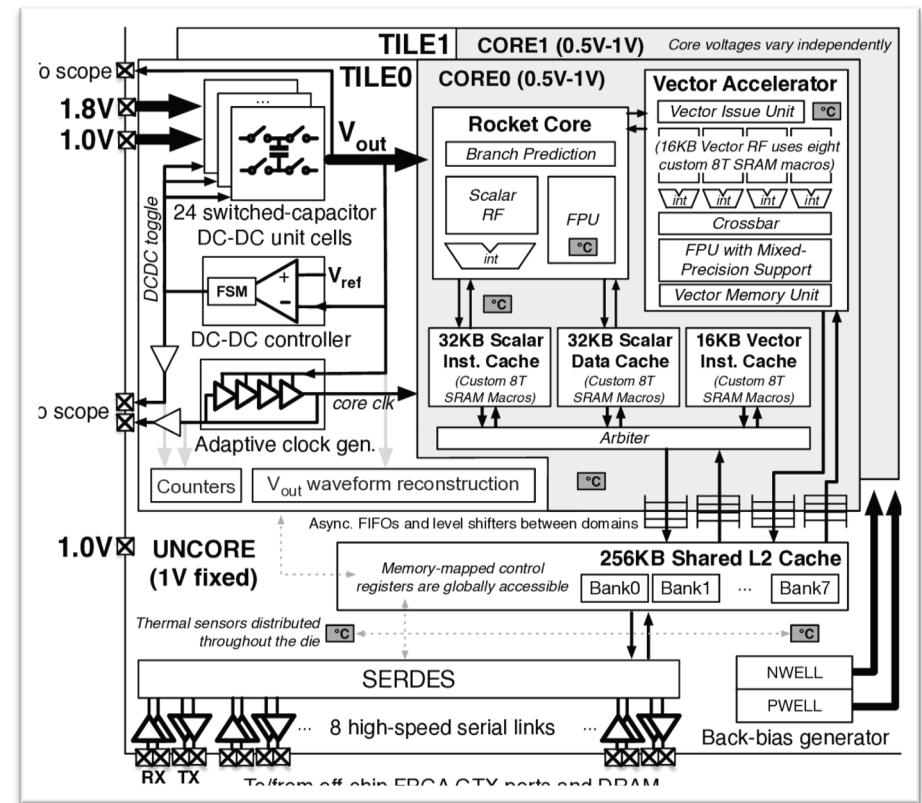
Generating Varied SoCs

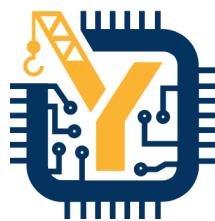


In industry: **SiFive Freedom E310**

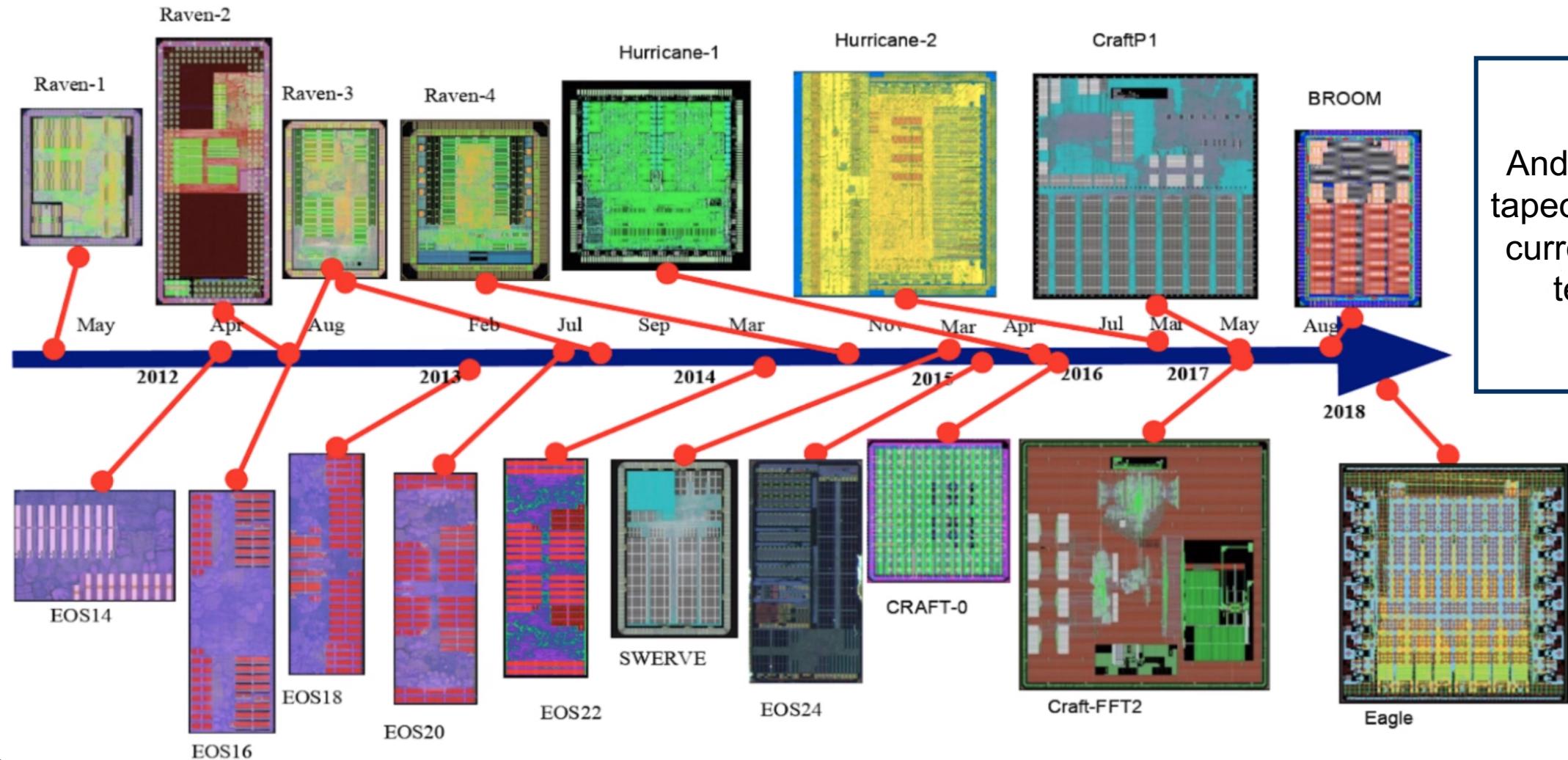


In academia: **UCB Hurricane-1**



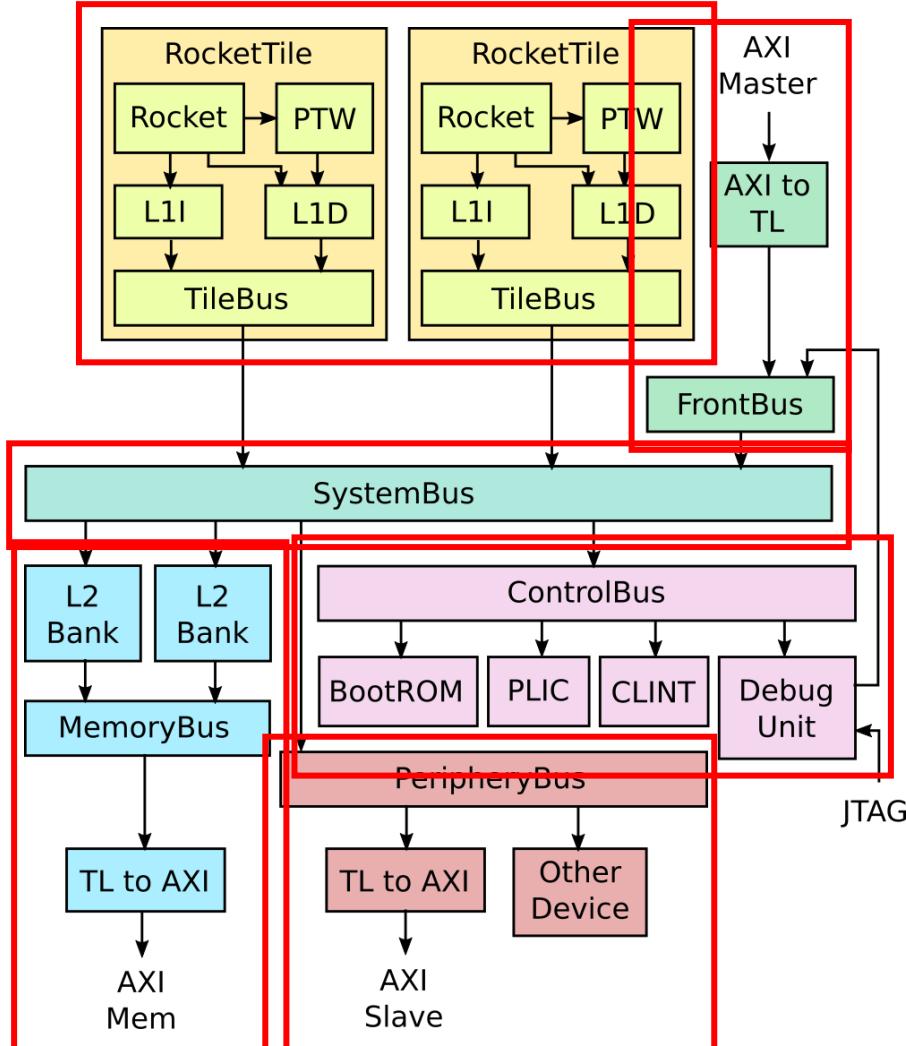


Used in Many Tapeouts





Structure of a Rocket Chip SoC



Tiles: unit of replication for a core

- CPU
- L1 Caches
- Page-table walker

L2 banks:

- Receive memory requests

FrontBus:

- Connects to DMA devices

ControlBus:

- Connects to core-complex devices

PeripheryBus:

- Connects to other devices

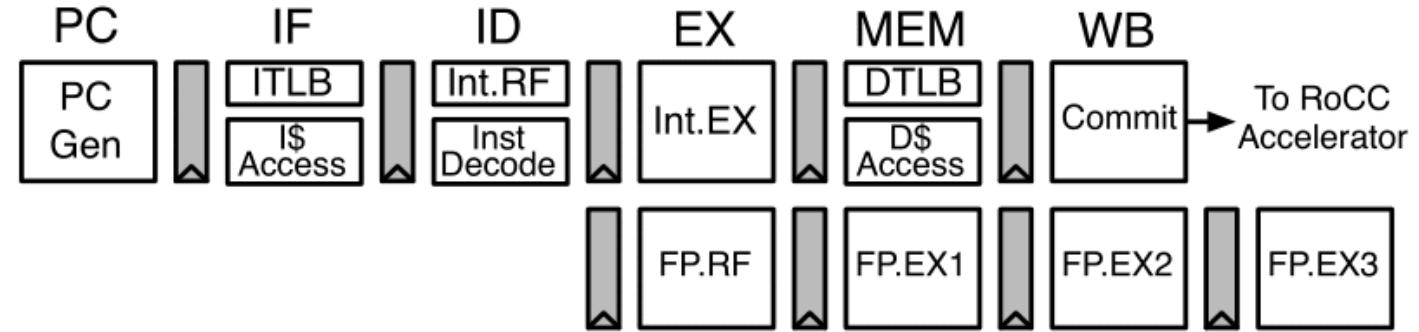
SystemBus:

- Ties everything together





The Rocket In-Order Core



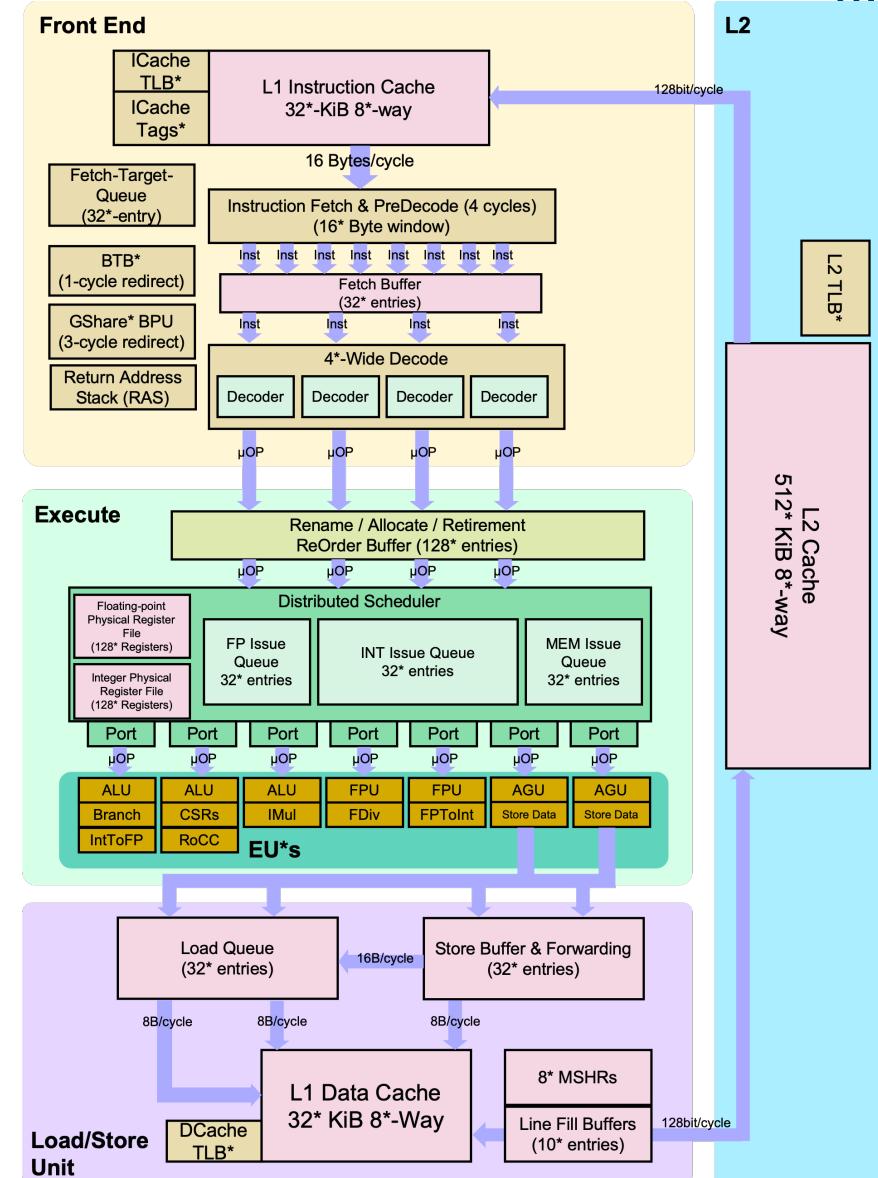
- First open-source RISC-V CPU
- In-order, single-issue RV64GC core
 - Floating-point via Berkeley hardfloat library
 - RISC-V Compressed
 - Physical Memory Protection (PMP) standard
 - Supervisor ISA and Virtual Memory
- Boots Linux
- Supports Rocket Chip Coprocessor (RoCC) interface
- L1 I\$ and D\$
 - Caches can be configured as scratchpads



BOOM: The Berkeley Out-of-Order Machine



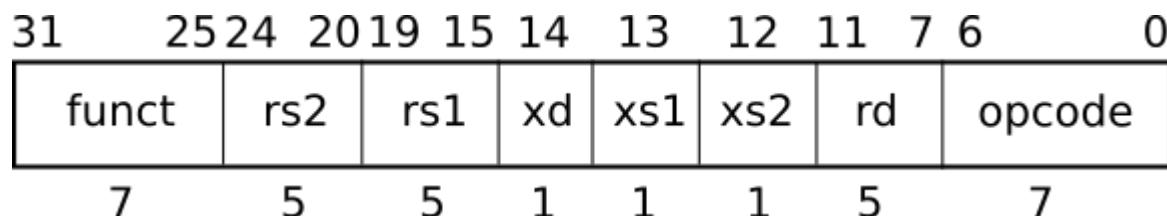
- Superscalar RISC-V OoO core
- Fully integrated in Rocket Chip ecosystem
- Open-source and described in Chisel
- Parameterizable generator
- Taped-out ([BROOM](#) at HC18, BEAGLE '21)
- Full RV64GC ISA support
 - FP, RVC, Atomics, PMPs, VM, Breakpoints, RoCC
 - Runs real OS's, software
- Drop-in replacement for Rocket



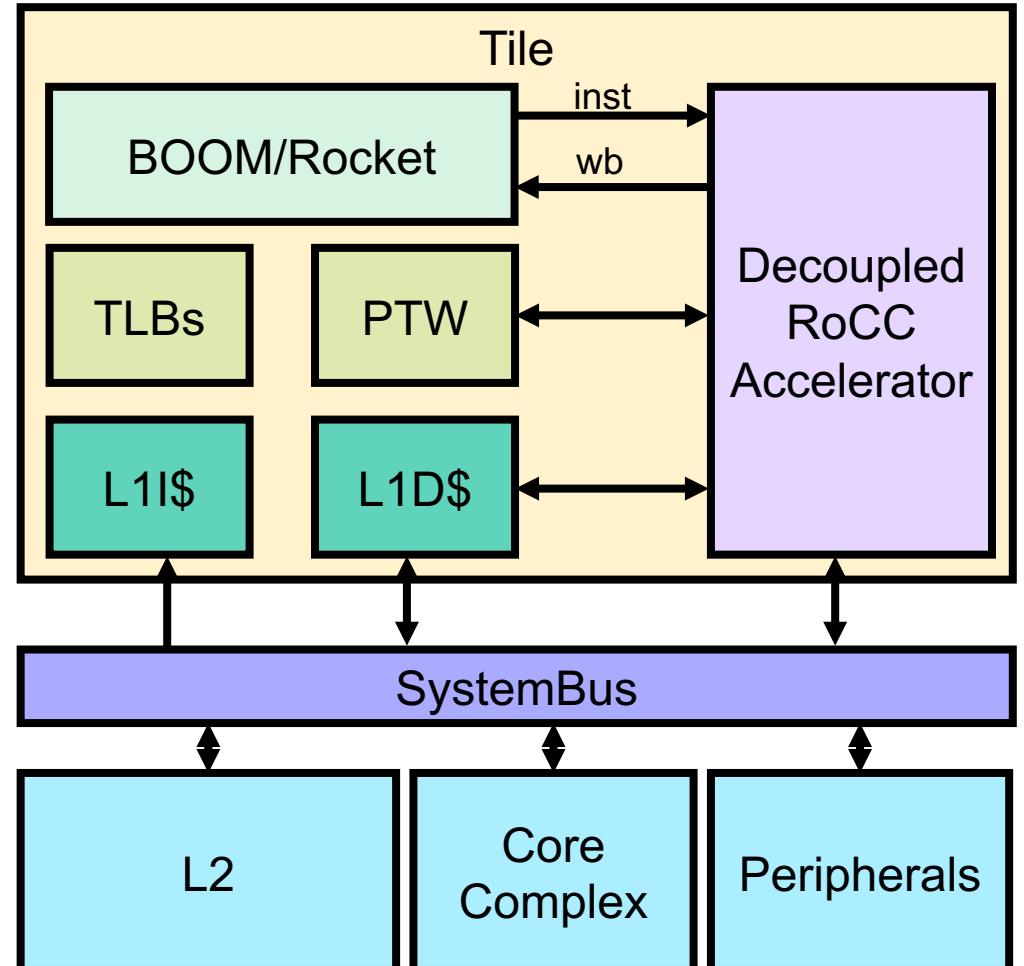
RoCC Accelerators



- **RoCC:** Rocket Chip Coprocessor
- Execute custom RISC-V instructions for a custom extension



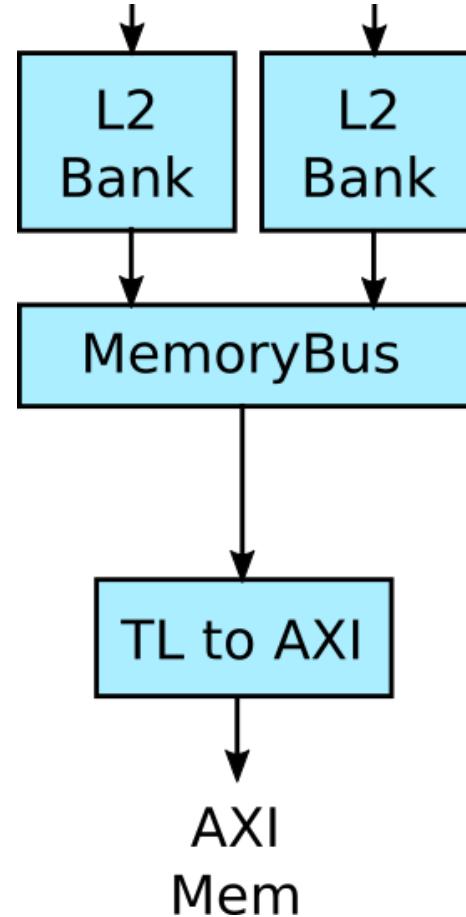
- Examples of RoCC accelerators
 - Vector accelerators
 - Memcpy accelerator
 - Machine-learning accelerators
 - Java GC accelerator

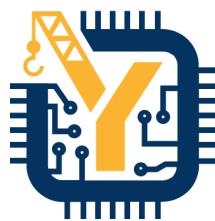




L2 Cache and Memory System

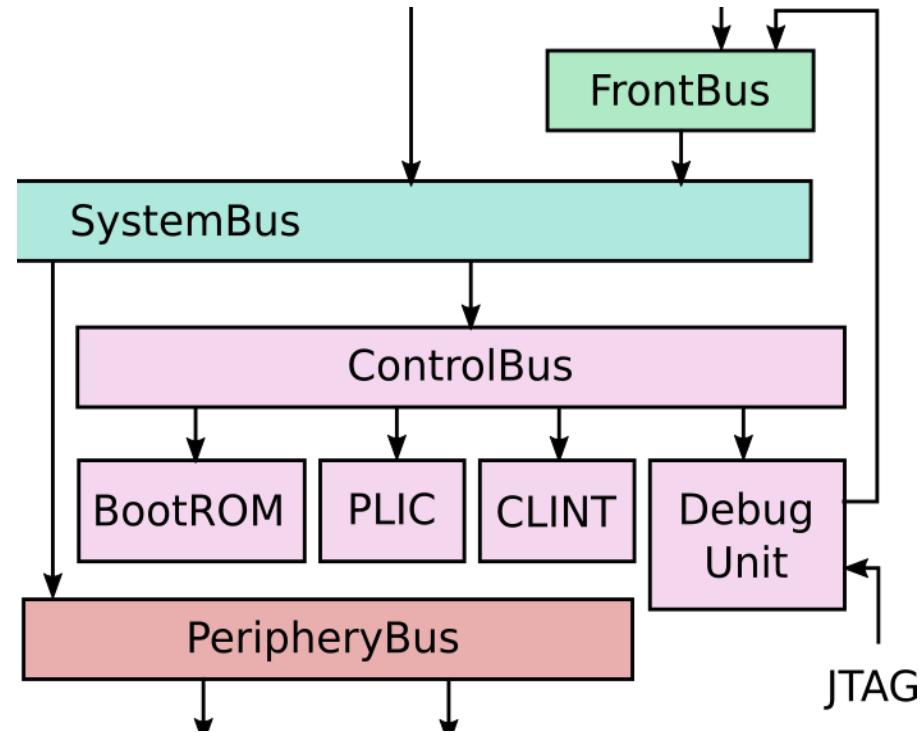
- Multi-bank shared L2
 - SiFive's open-source IP
 - Fully coherent
 - Configurable size, associativity
 - Supports atomics, prefetch hints
- Non-caching L2 Broadcast Hub
 - Coherence w/o caching
 - Bufferless design
- Multi-channel memory system
 - Conversion to AXI4 for compatible DRAM controllers

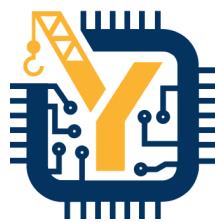




Core Complex Devices

- BootROM
 - First-stage bootloader
 - DeviceTree
- PLIC
- CLINT
 - Software interrupts
 - Timer interrupts
- Debug Unit
 - DMI
 - JTAG



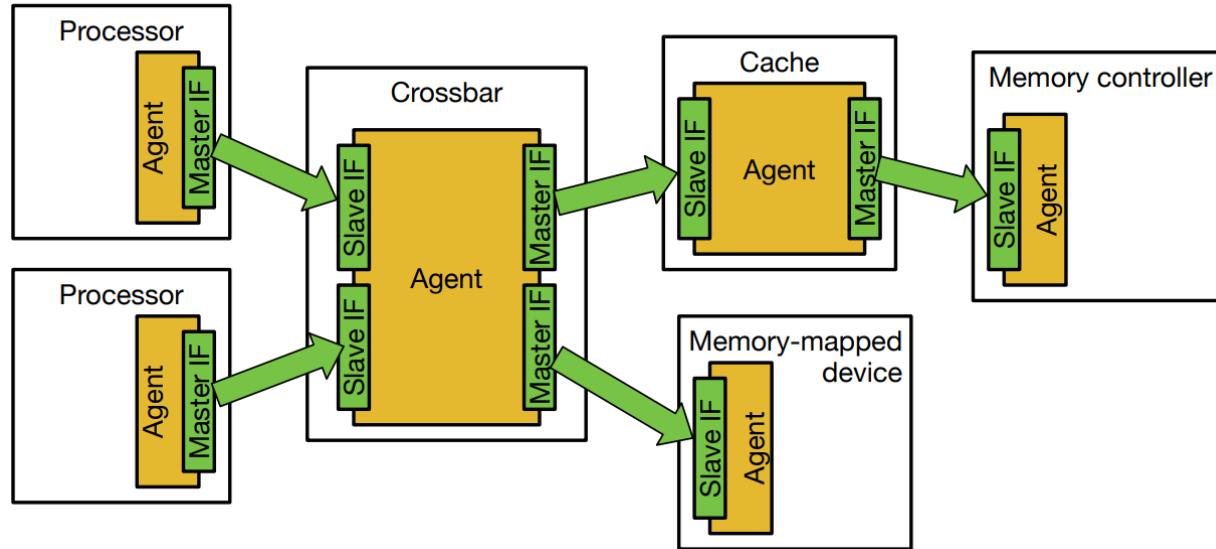


Other Chipyard Blocks

- **Hardfloat:** Parameterized Chisel generators for hardware floating-point units
- **IceNet:** Custom NIC for FireSim simulations
- **SiFive-Blocks:** Open-sourced Chisel peripherals
 - GPIO, SPI, UART, etc.
- **TestchipIP:** Berkeley utilities for chip testing/bringup
 - Tethered serial interface
 - Simulated block device
- **Hwacha:** Decoupled vector-fetch RoCC accelerator
- **Gemmini:** Systolic-array matrix multiplication RoCC accelerator
- **SHA3:** Educational SHA3 RoCC accelerator



TileLink Interconnect



- Free and open chip-scale interconnect standard
- Supports multiprocessors, coprocessors, accelerators, DMA, peripherals, etc.
- Provides a physically addressed, shared-memory system
- Supports cache-coherent shared memory, MOESI-equivalent protocol
- Verifiable deadlock freedom for conforming SoCs



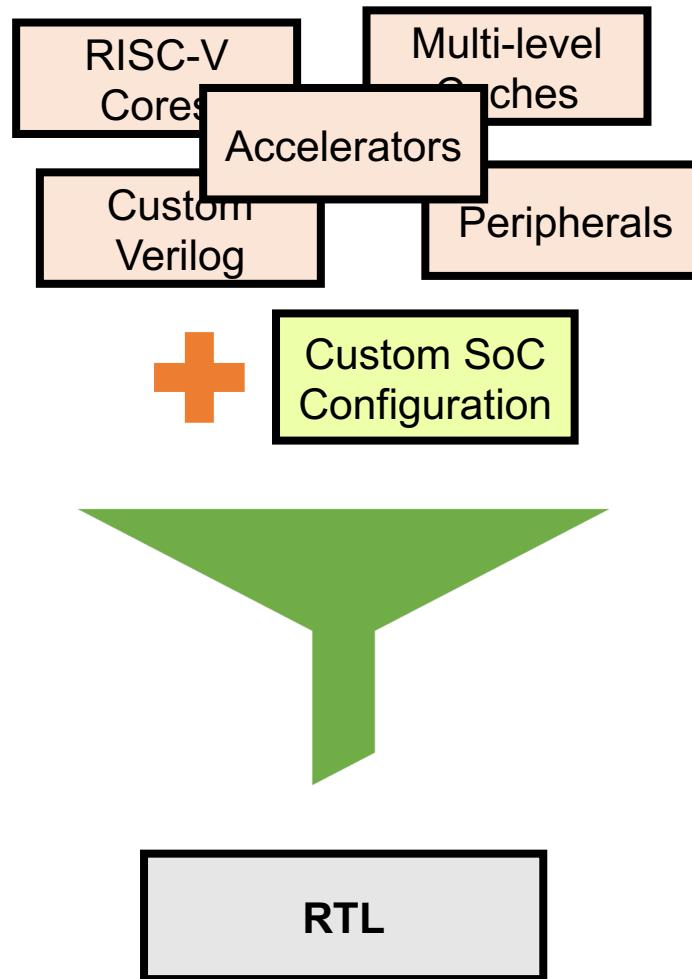
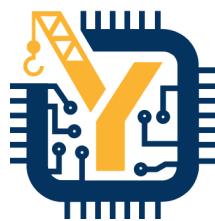
TileLink Interconnect



- Three different protocol levels with increasing complexity
 - TL-UL (Uncached Lightweight)
 - TL-UH (Uncached Heavyweight)
 - TL-C (Cached)
- Rocket Chip provides library of reusable TileLink widgets
 - Conversion to/from AXI4, AHB, APB
 - Conversion among TL-UL, TL-UH, TL-C
 - Crossbar generator
 - Width / logical size converters
 - TLMonitor conformance checker



Integration



Challenges of configuring generators

- How do we allow parameterization of interdependent generators?
 - The memory system is a function of the number of cores, cache size, DRAM size, MMIO addresses, etc.
 - **Solution: Diplomatic parameter negotiation**
- How do we flexibly and reusably describe a system's parameterization?
 - Enable rapid design-space exploration
 - **Solution: Context-dependent parameterization (aka Rocketchip Configs)**



Diplomacy



Problem: Interconnects are difficult to parameterize correctly

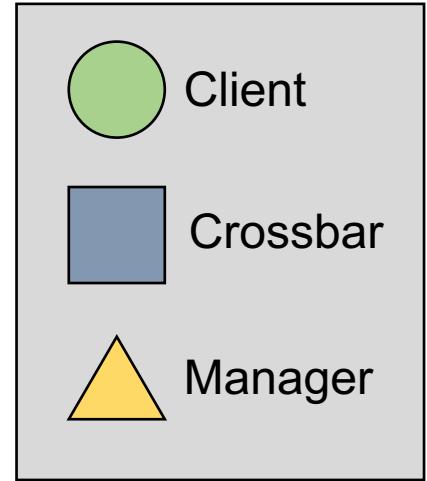
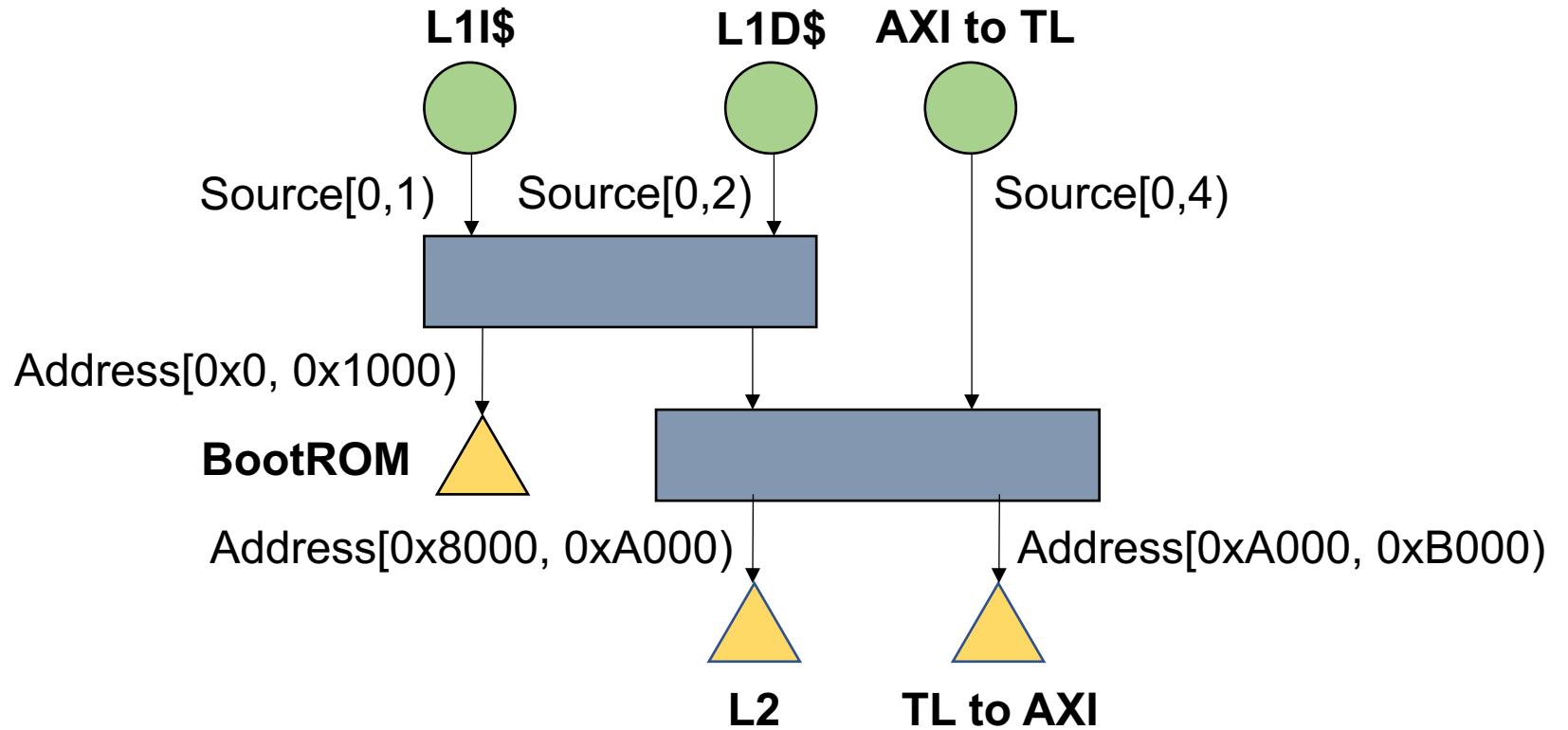
- Complex interconnect graph with many nodes
- Nodes are independently parameterized (but are interdependent)

Diplomacy: Framework for negotiating parameters between Chisel generators

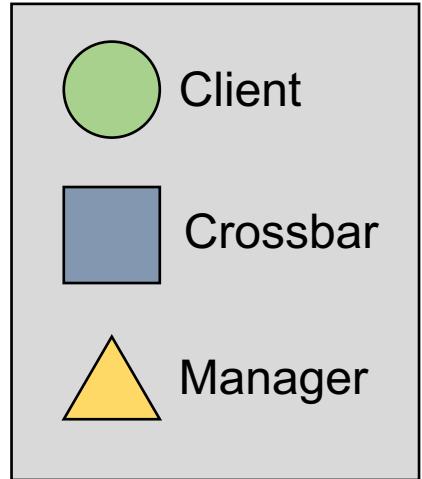
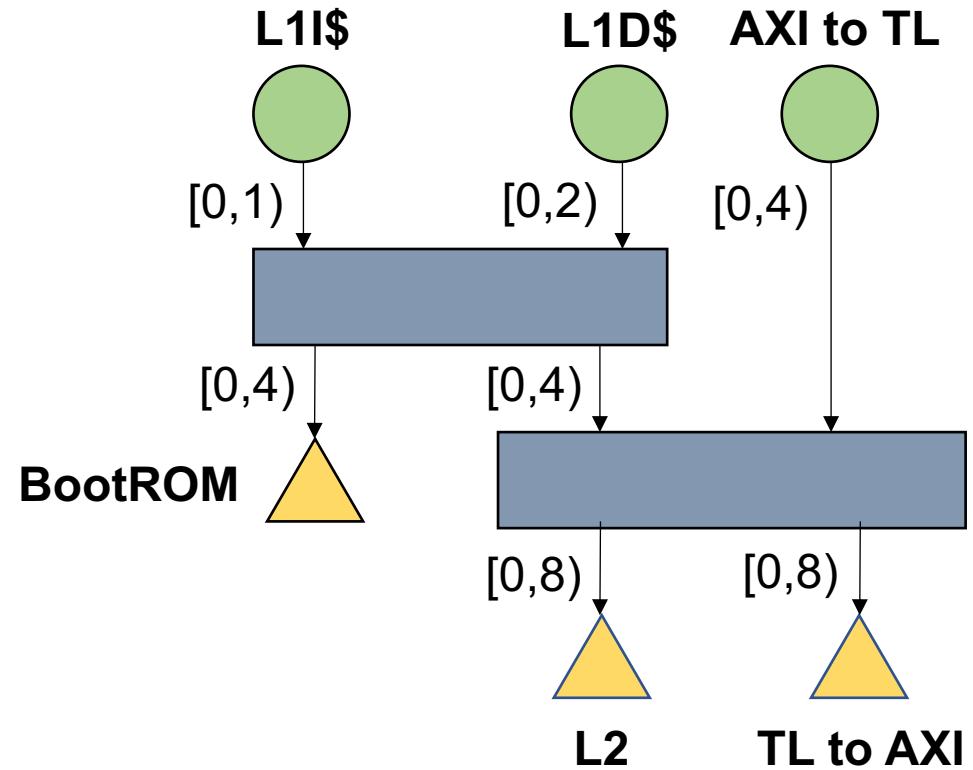
- Graphical abstraction of interconnectivity
- Diplomatic lazy modules follow two-phase elaboration
 - **Phase one:** nodes exchange configuration information with each other and decide final parameters
 - **Phase two:** Chisel RTL elaborates using calculated parameters
- Used extensively by RocketChip TileLink generators



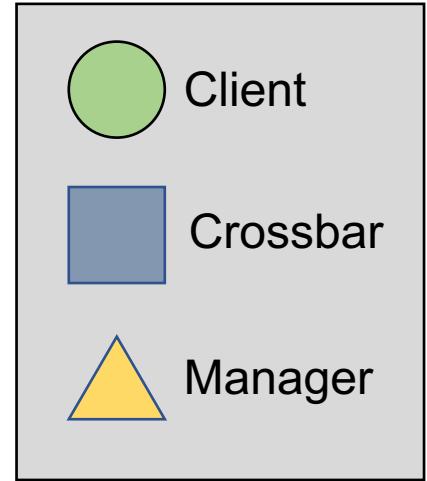
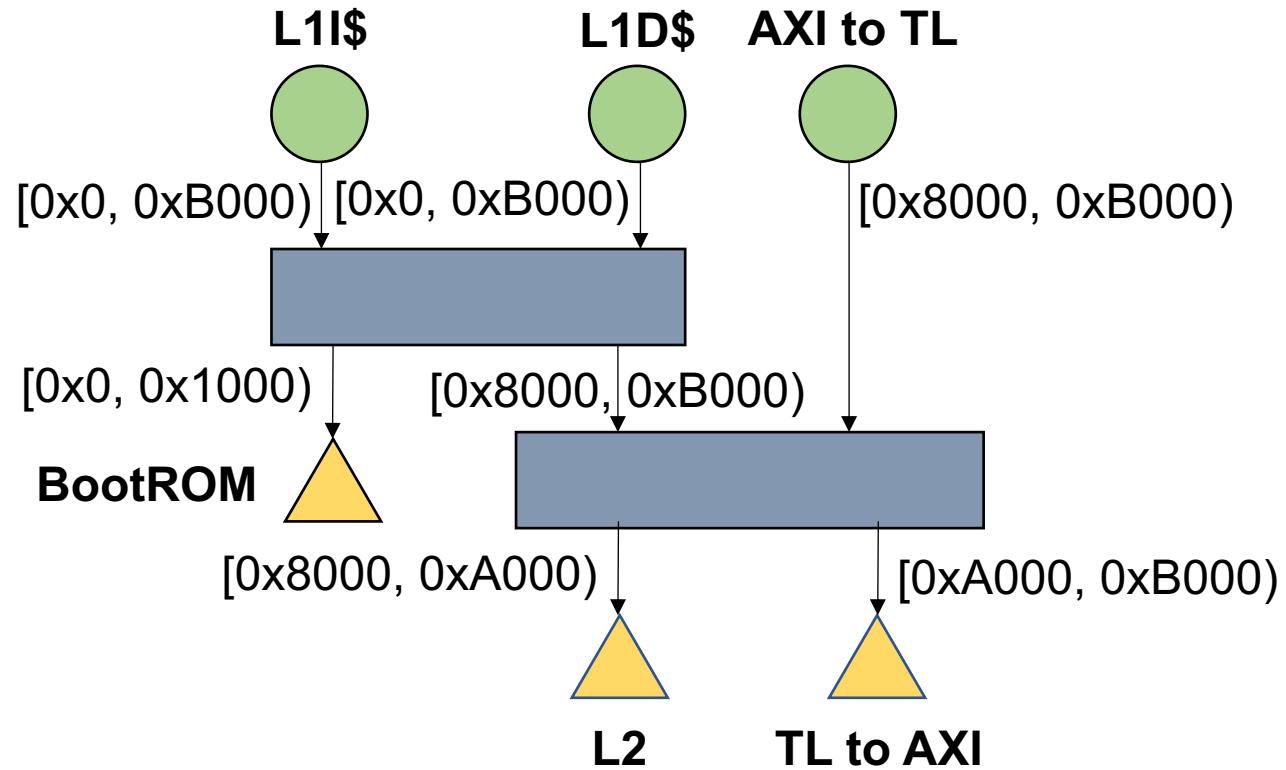
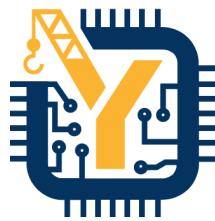
Diplomacy Example



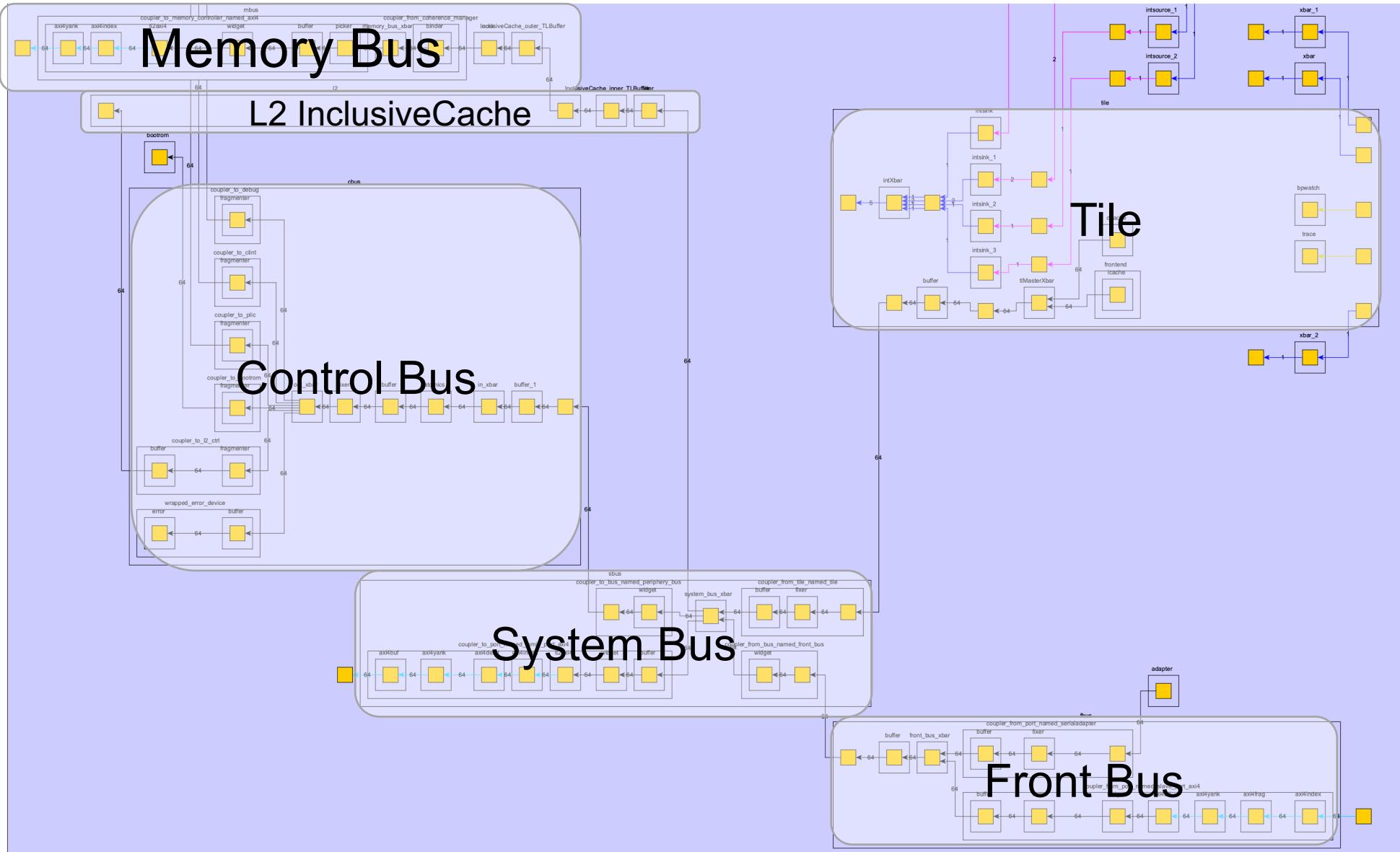
Diplomacy Example



Diplomacy Example



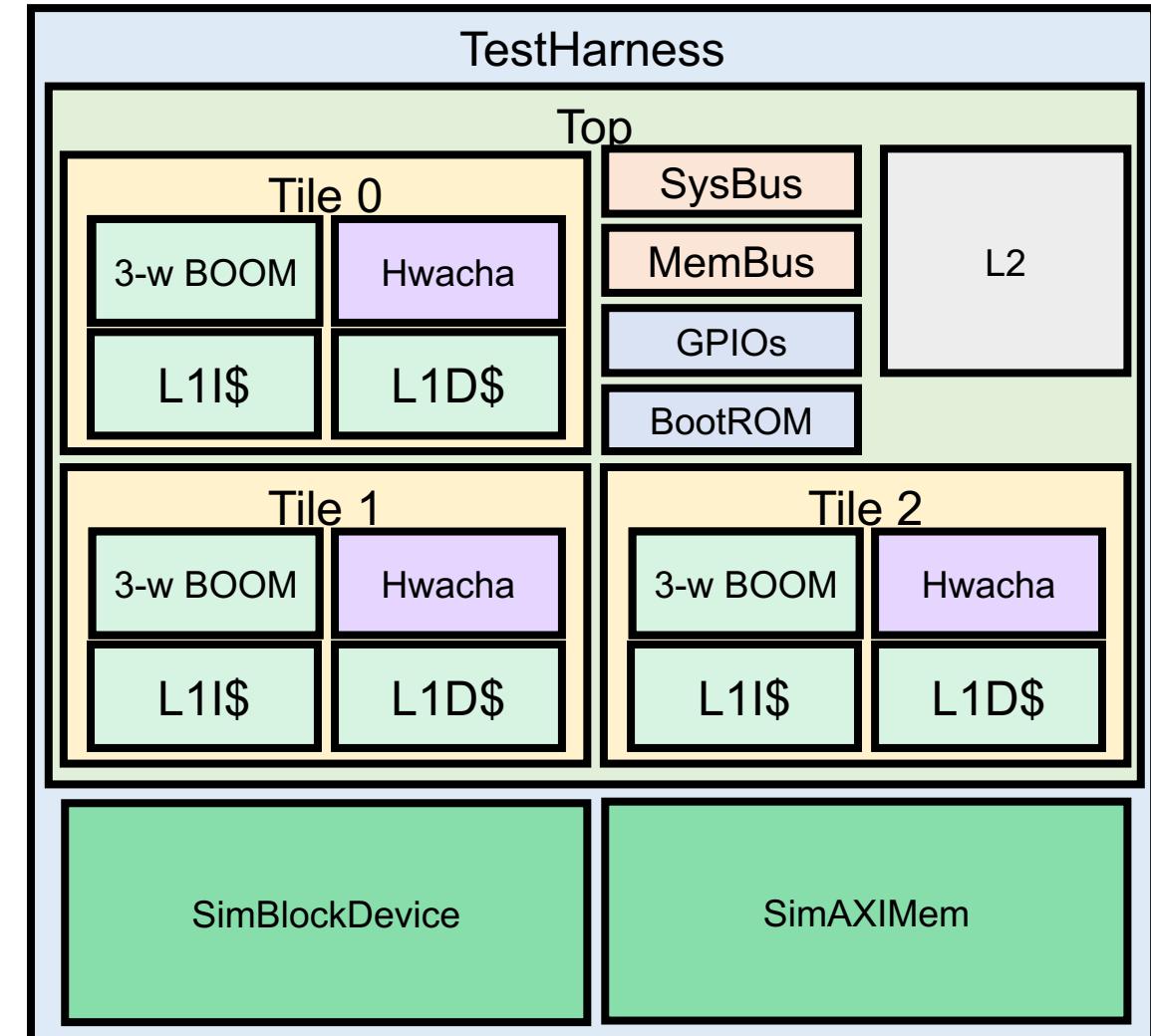
Diplomacy-generated Graph



Rocket Chip Configuration



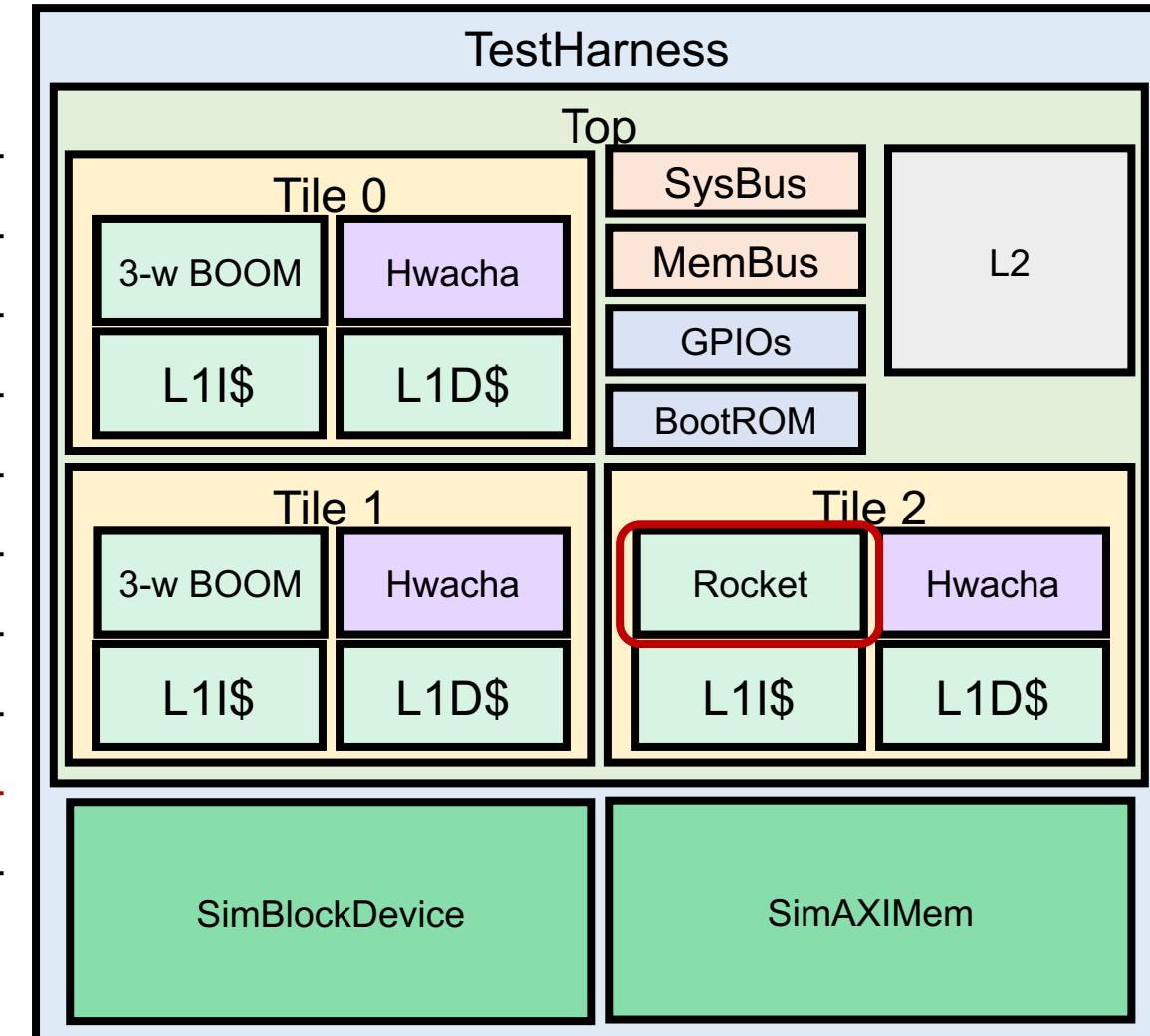
```
class MyCustomConfig extends Config(
    new WithExtMemSize((1<<30) * 2L)
    new WithBlockDevice
    new WithGPIO
    new WithBootROM
    new hwacha.DefaultHwachaConfig
    new WithInclusiveCache(capacityKB=1024)
    new boom.common.WithLargeBooms
    new boom.system.WithNBoomCores(3)
    new WithNormalBoomRocketTop
    new rocketchip.system.BaseConfig)
```





Rocket Chip Configuration

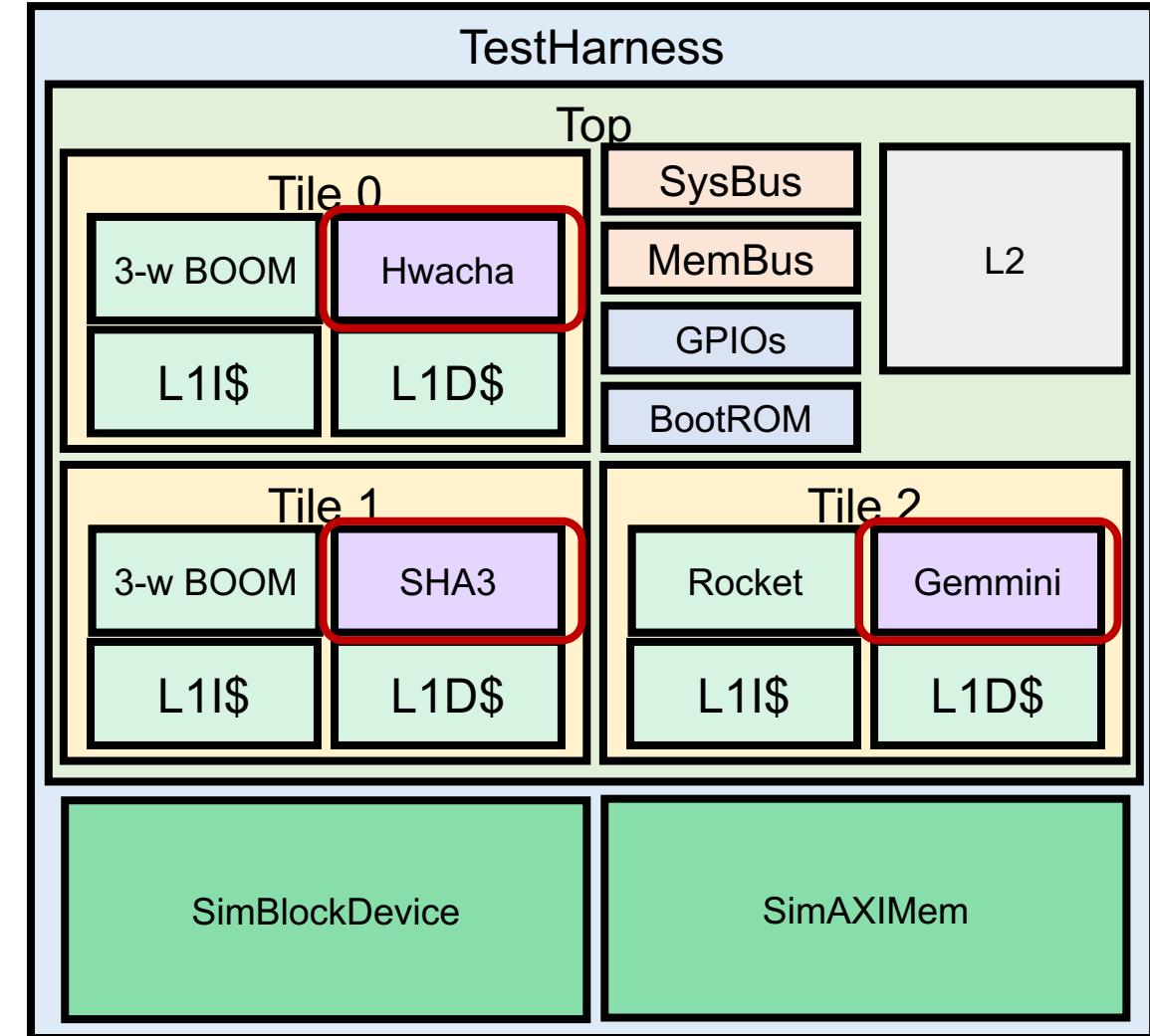
```
class MyCustomConfig extends Config  
    new WithExtMemSize((1<<30) * 2L)  
    new WithBlockDevice  
    new WithGPIO  
    new WithBootROM  
    new hwacha.DefaultHwachaConfig  
    new WithInclusiveCache(capacityKB=1024)  
    new boom.common.WithLargeBooms  
    new boom.system.WithNBoomCores(2)  
    new rocketchip.subsystem.WithNBigCores(1)  
    new WithNormalBoomRocketTop  
    new rocketchip.system.BaseConfig)
```

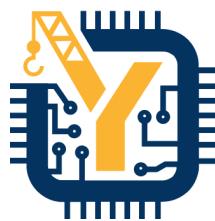




Rocket Chip Configuration

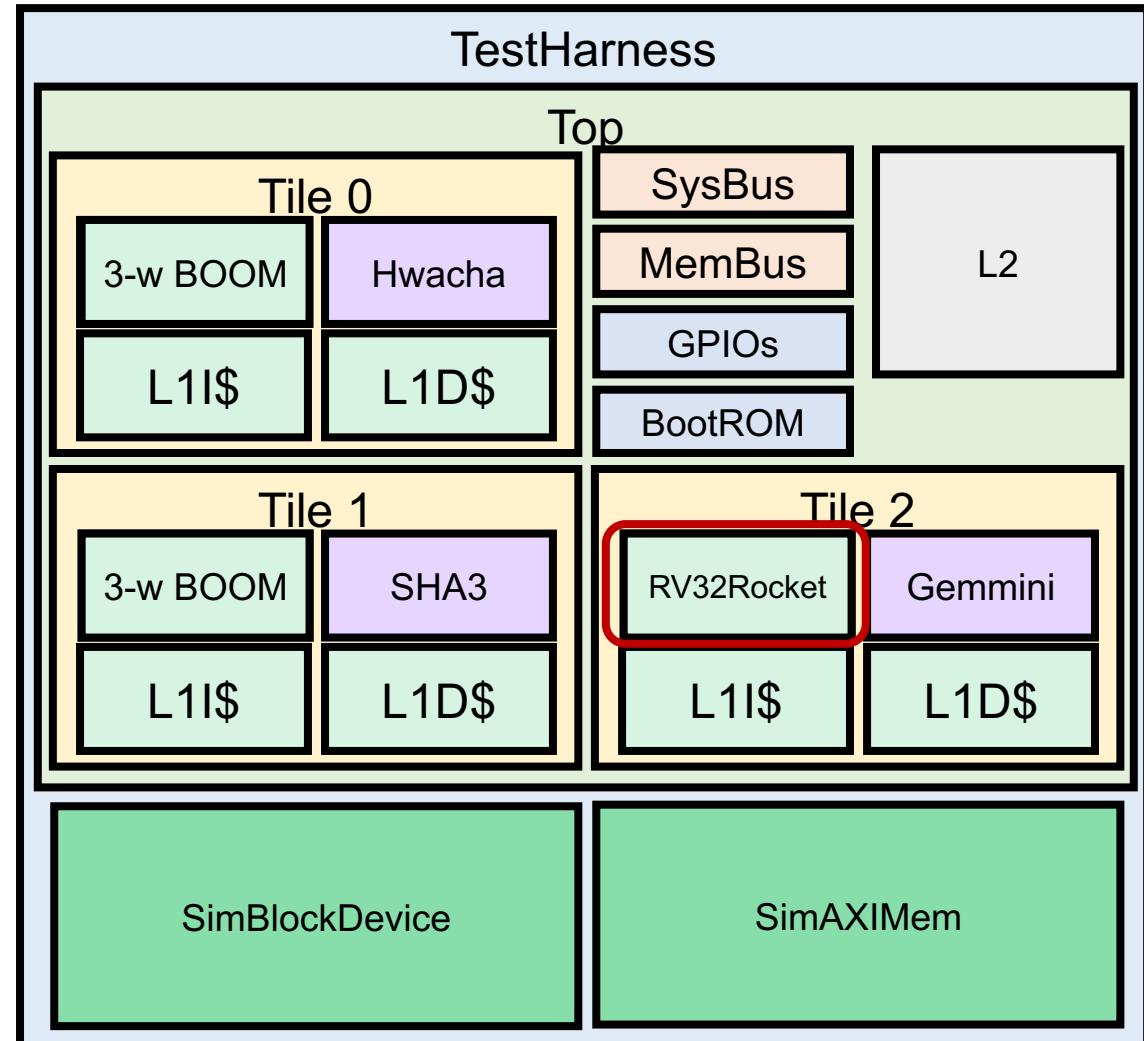
```
class MyCustomConfig extends Config(
    new WithExtMemSize((1<<30) * 2L)
    new WithBlockDevice
    new WithGPIO
    new WithBootROM
    new WithMultiRoCCGemmini(2)
    new WithMultiRoCCSha3(1)
    new WithMultiRoCCHwacha(0)
    new WithInclusiveCache(capacityKB=1024)
    new boom.common.WithLargeBooms
    new boom.system.WithNBoomCores(2)
    new rocketchip.subsystem.WithNBigCores(1)
    new WithNormalBoomRocketTop
    new rocketchip.system.BaseConfig)
```





Rocket Chip Configuration

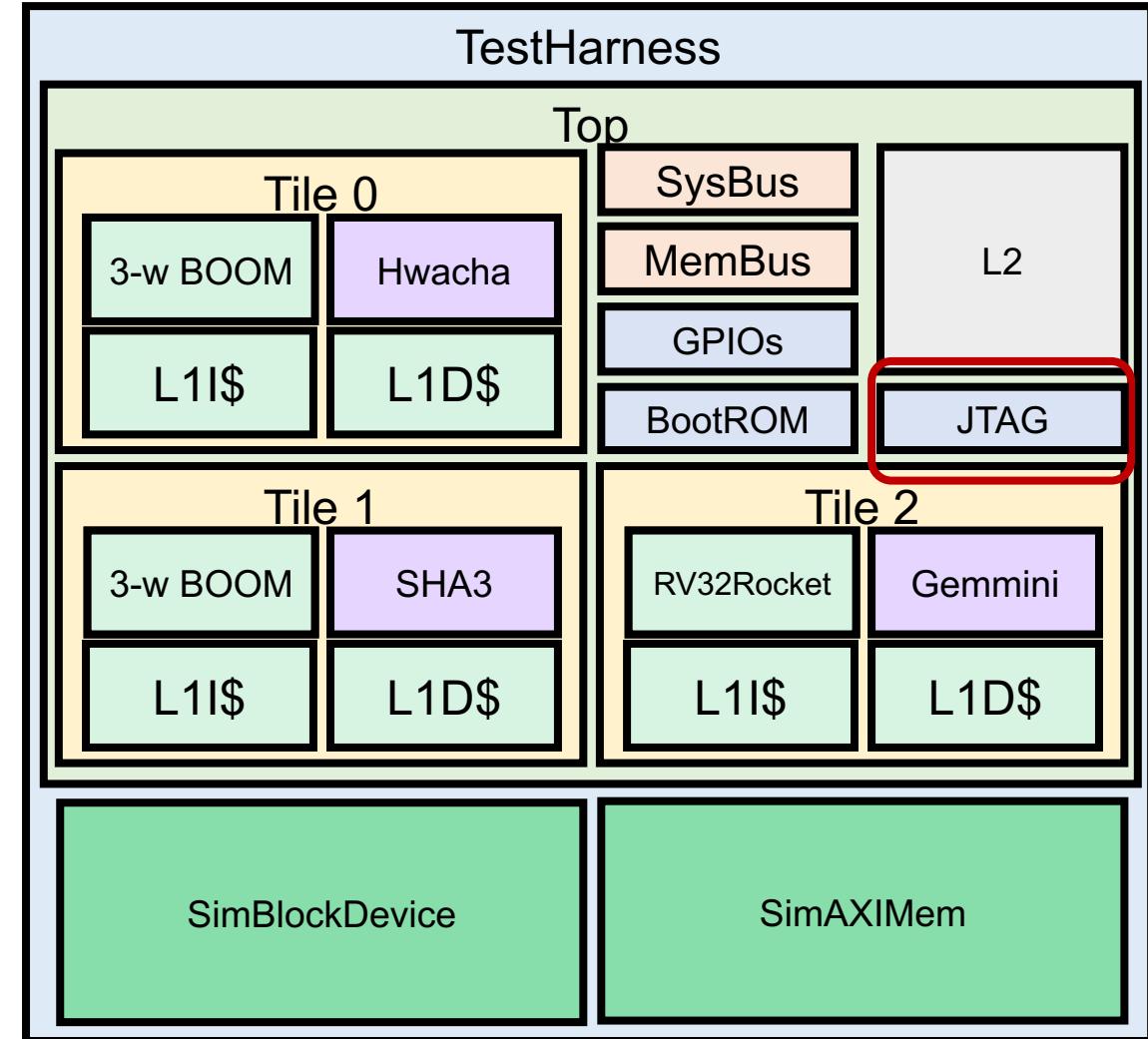
```
class MyCustomConfig extends Config(
    new WithExtMemSize((1<<30) * 2L)
    new WithBlockDevice
    new WithGPIO
    new WithBootROM
    new WithMultiRoCCGemmini(2)
    new WithMultiRoCCSha3(1)
    new WithMultiRoCCHwacha(0)
    new WithInclusiveCache(capacityKB=1024)
    new boom.common.WithLargeBooms
    new boom.system.WithNBoomCores(2)
    new rocketchip.subsystem.WithRV32
    new rocketchip.subsystem.WithNBigCores(1)
    new WithNormalBoomRocketTop
    new rocketchip.system.BaseConfig)
```



Rocket Chip Configuration



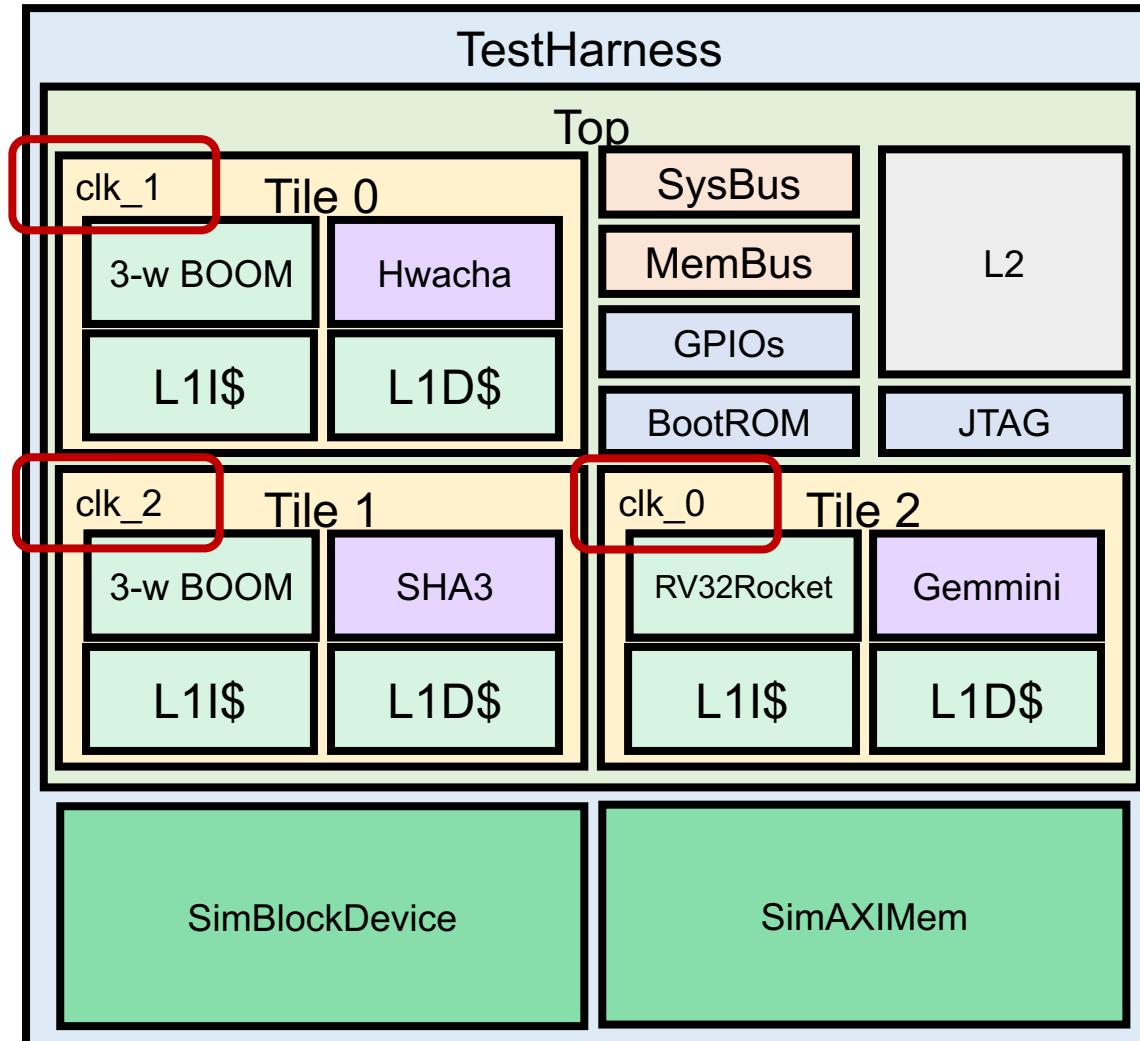
```
class MyCustomConfig extends Config
    new WithExtMemSize((1<<30) * 2L)          ++
    new WithBlockDevice                          ++
    new WithGPIO                                ++
    new WithJtagDTM                            ++
    new WithBootROM                            ++
    new WithMultiRoCCGemmini(2)                 ++
    new WithMultiRoCCSha3(1)                     ++
    new WithMultiRoCCHwacha(0)                  ++
    new WithInclusiveCache(capacityKB=1024)    ++
    new boom.common.WithLargeBooms            ++
    new boom.system.WithNBoomCores(2)           ++
    new rocketchip.subsystem.WithRV32          ++
    new rocketchip.subsystem.WithNBigCores(1) ++
    new WithNormalBoomRocketTop               ++
    new rocketchip.system.BaseConfig)
```



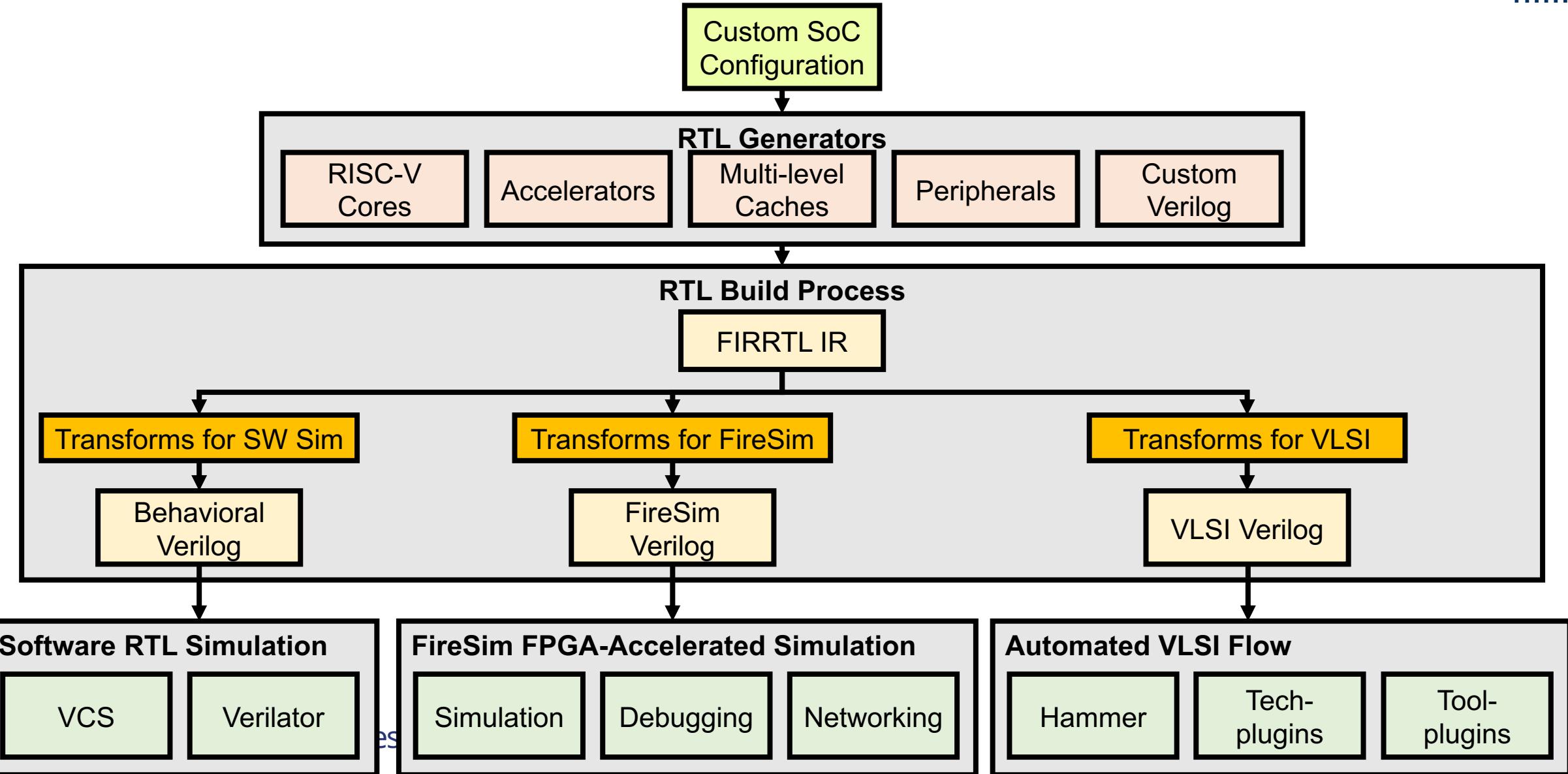


Rocket Chip Configuration

```
class MyCustomConfig extends Config(
    new WithExtMemSize((1<<30) * 2L)          ++
    new WithBlockDevice                         ++
    new WithGPIO                                ++
    new WithJtagDTM                            ++
    new WithBootROM                            ++
    new WithRationalBoomTiles                   ++
    new WithRationalRocketTiles                 ++
    new WithMultiRoCCGemmini(2)                  ++
    new WithMultiRoCCSha3(1)                     ++
    new WithMultiRoCCHwacha(0)                   ++
    new WithInclusiveCache(capacityKB=1024)     ++
    new boom.common.WithLargeBooms              ++
    new boom.system.WithNBoomCores(2)             ++
    new rocketchip.subsystem.WithRV32            ++
    new rocketchip.subsystem.WithNBigCores(1)    ++
    new WithNormalBoomRocketTop                ++
    new rocketchip.system.BaseConfig)
```



Chipyard Unified Flows





FireSim

Scalable FPGA-Accelerated
Cycle-Accurate Hardware
Simulation in the Cloud

<https://fires.im>

 @firesimproject

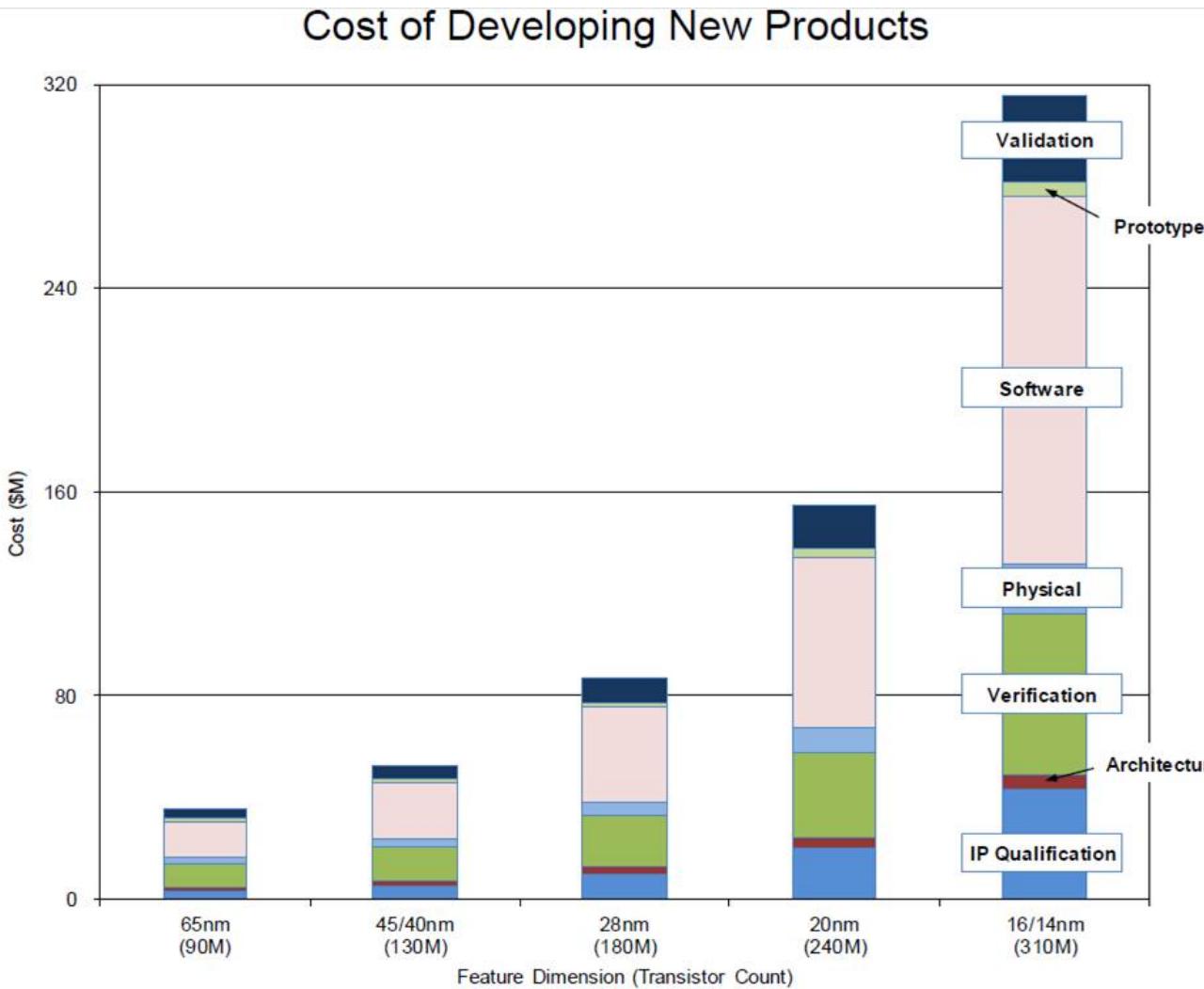
Sagar Karandikar, David Biancolin, Howard Mao, Alon Amid, Nathan Pemberton, Albert Magyar, Albert Ou, Randy Katz, Borivoje Nikolić, Jonathan Bachrach, Krste Asanović



Berkeley Architecture Research



The Non Recurring Engineering (NRE) Cost Barrier



- NRE is a huge barrier to building chips
- Many sources; death by a thousand cuts
→ Requires a large joint effort

FireSim's focus:

Enable better *full-system* simulation for pre-silicon

- Verification
- Validation
- Software development





Want:

- As fast as silicon
- As detailed as silicon
- All the benefits of SW-based simulator
- Low cost

Reality: can only pick ~2.5

Our Thesis:

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



Useful Trends Throughout the Stack

Open ISA



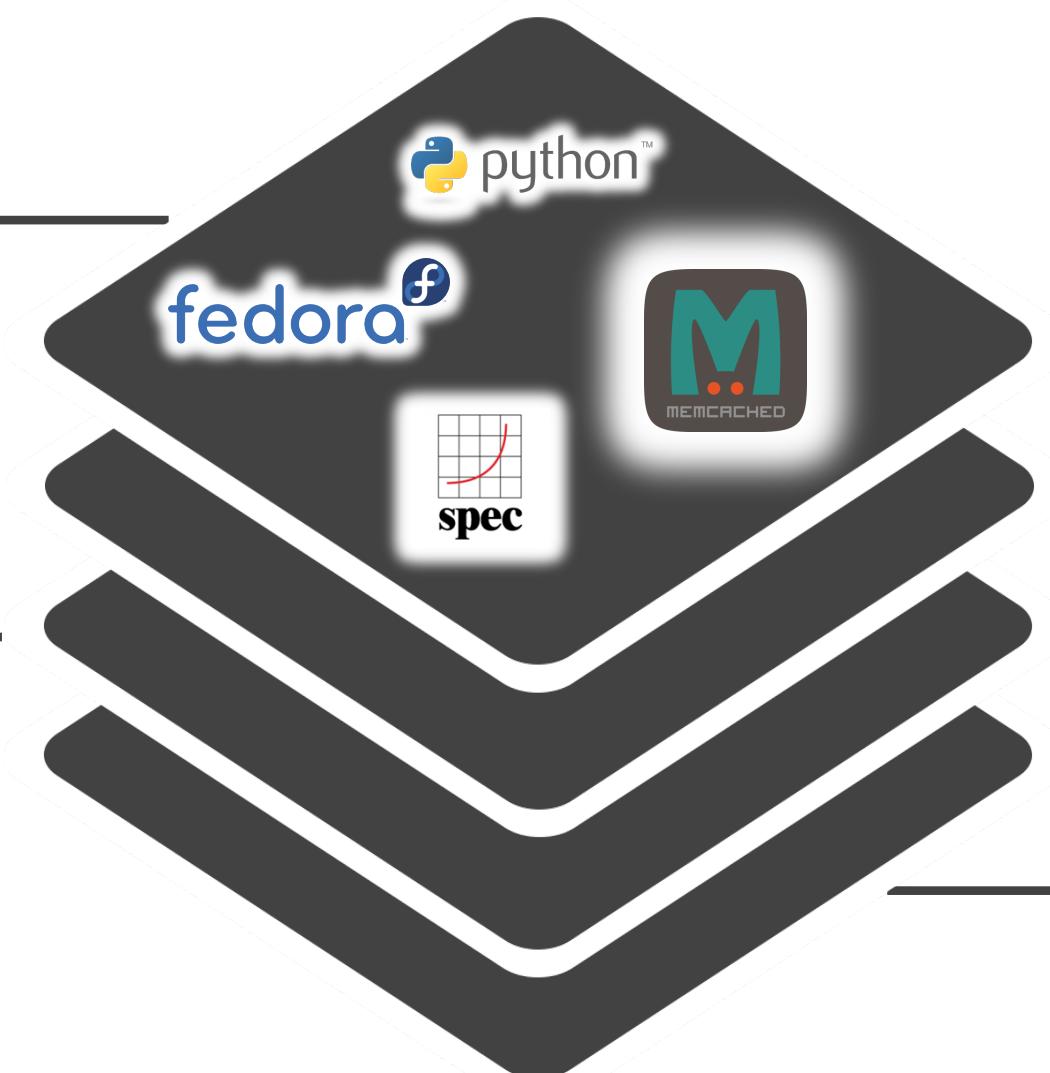
CHISEL

High-Productivity
Hardware Design

Language & IR



Berkeley Architecture Research



Open, Silicon-Proven
SoC Implementations



FPGAs in the Cloud





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*)
 - HW and/or SW IO models (e.g. UART, Ethernet, DRAM, etc.)
 - Workload descriptions
- Produces:
 - Fast, cycle-exact simulation of your design + models around it
 - Automatically deployed to cloud FPGAs (AWS EC2 F1)

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

[2] 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 FPGA emulation platforms
- 2) Uses cloud FPGAs
 - Inexpensive, elastic supply of large FPGAs
 - Easy to collaborate with other researchers
- 3) Open-source





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:

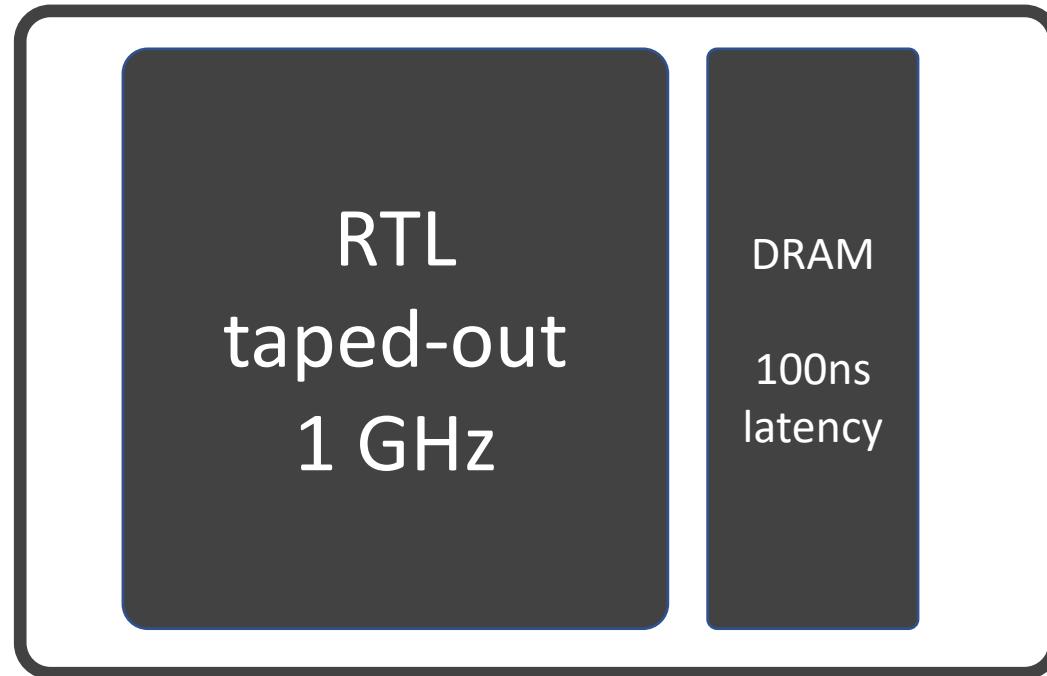
- 1) FPGA resources may not be an accurate model (ex. previous slide)
- 2) Simulations are non-deterministic
- 3) Different host FPGAs produce different simulation results





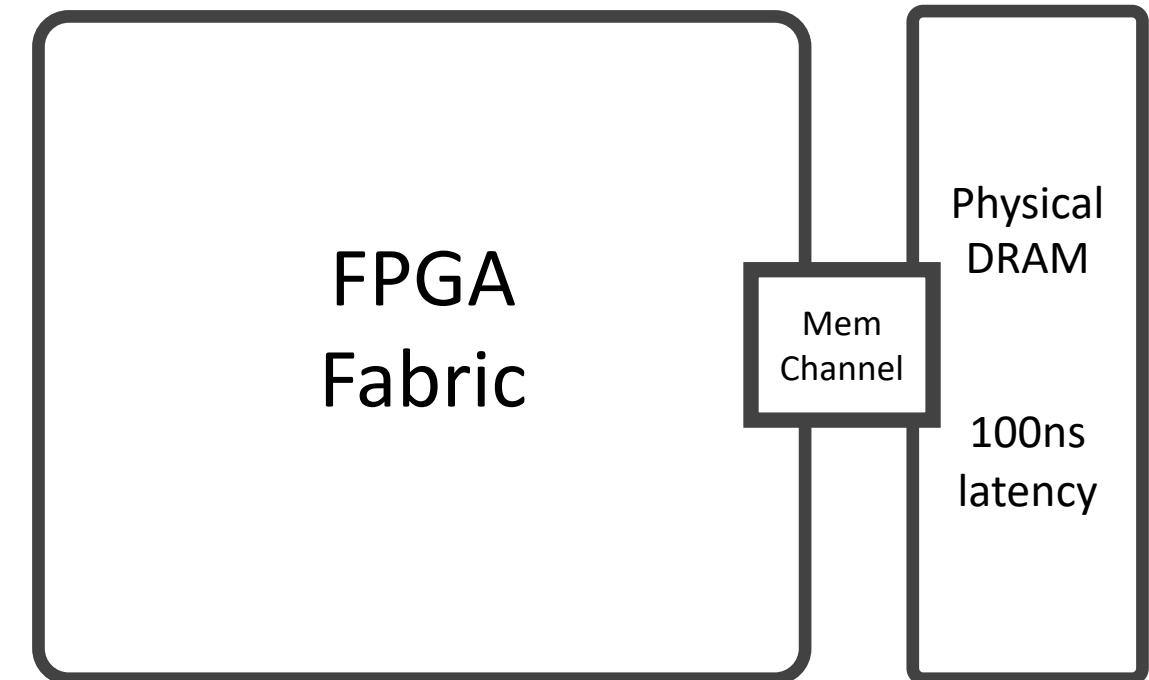
Separating Target and Host

Target: the machine under simulation



Closed simulation world.

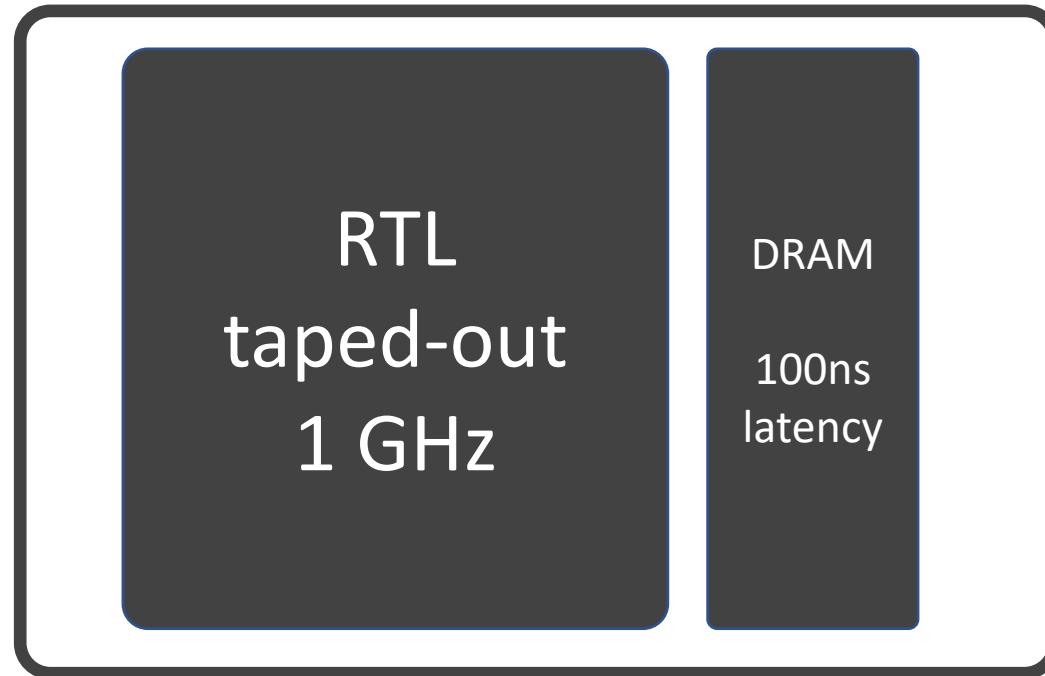
Host: the machine executing (*hosting*) the simulation





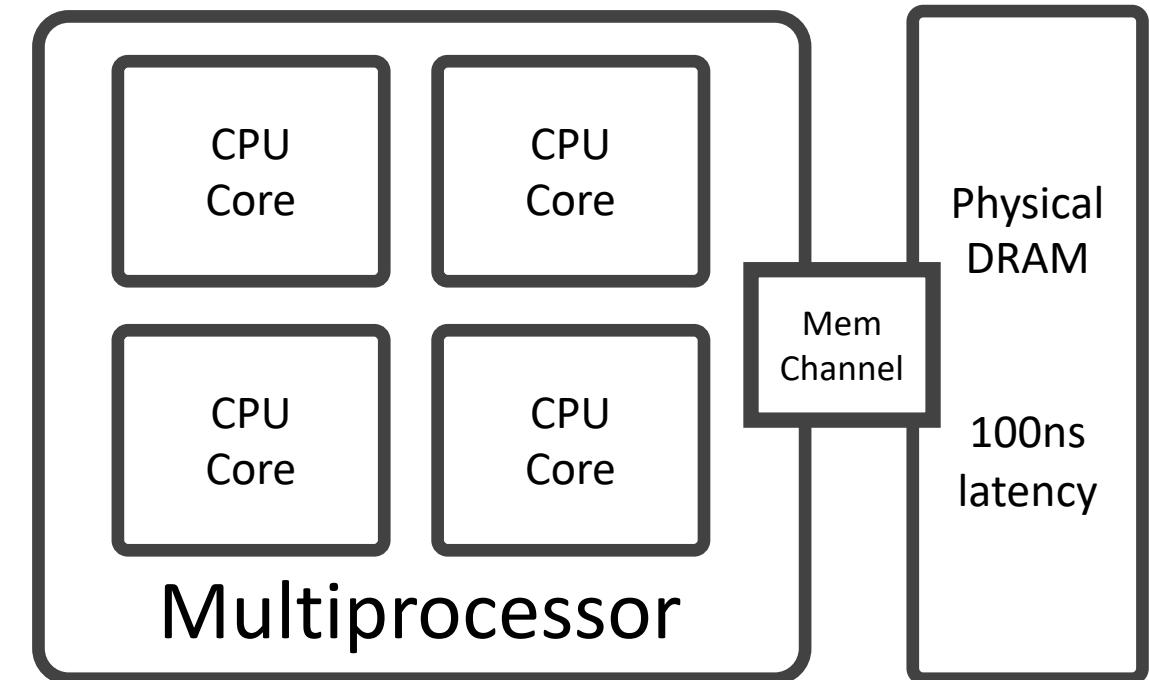
Separating Target and Host

Target: the machine under simulation



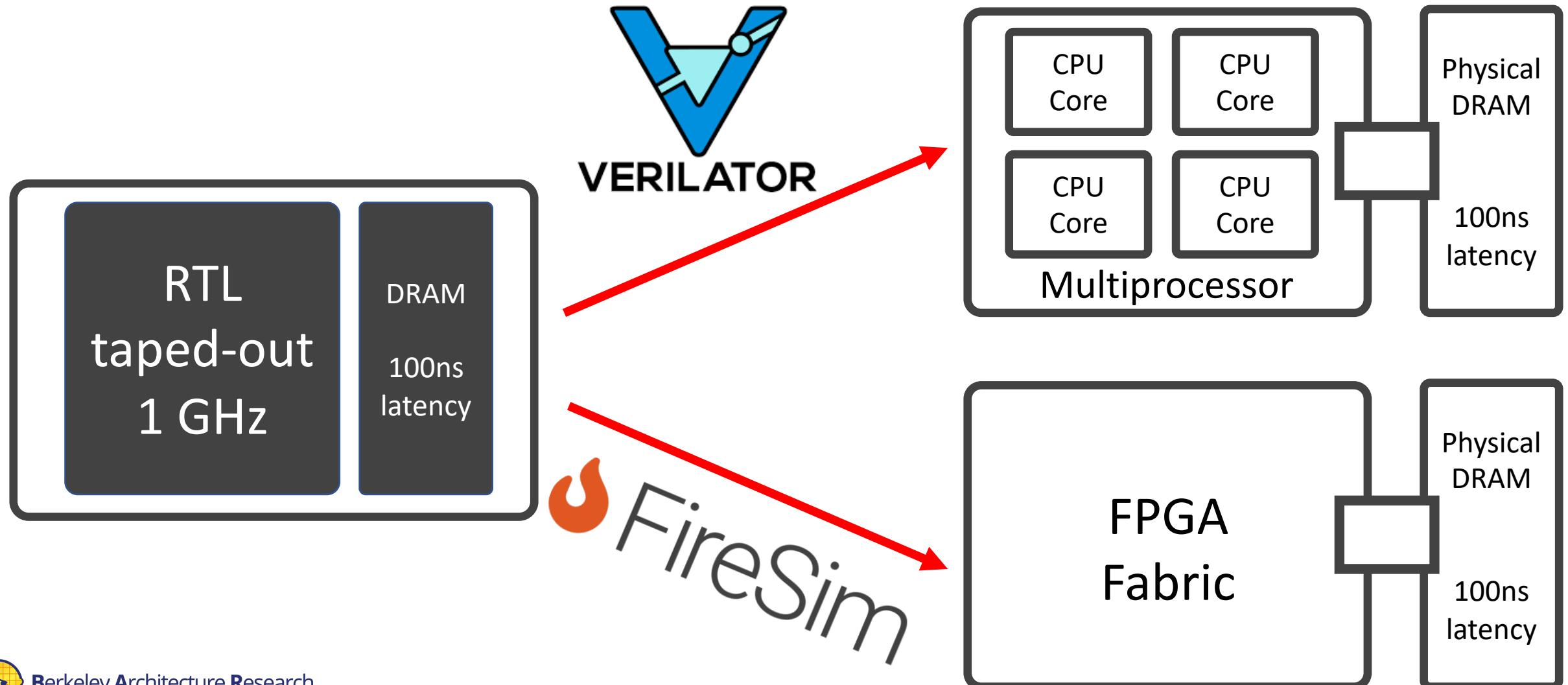
Closed simulation world.

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





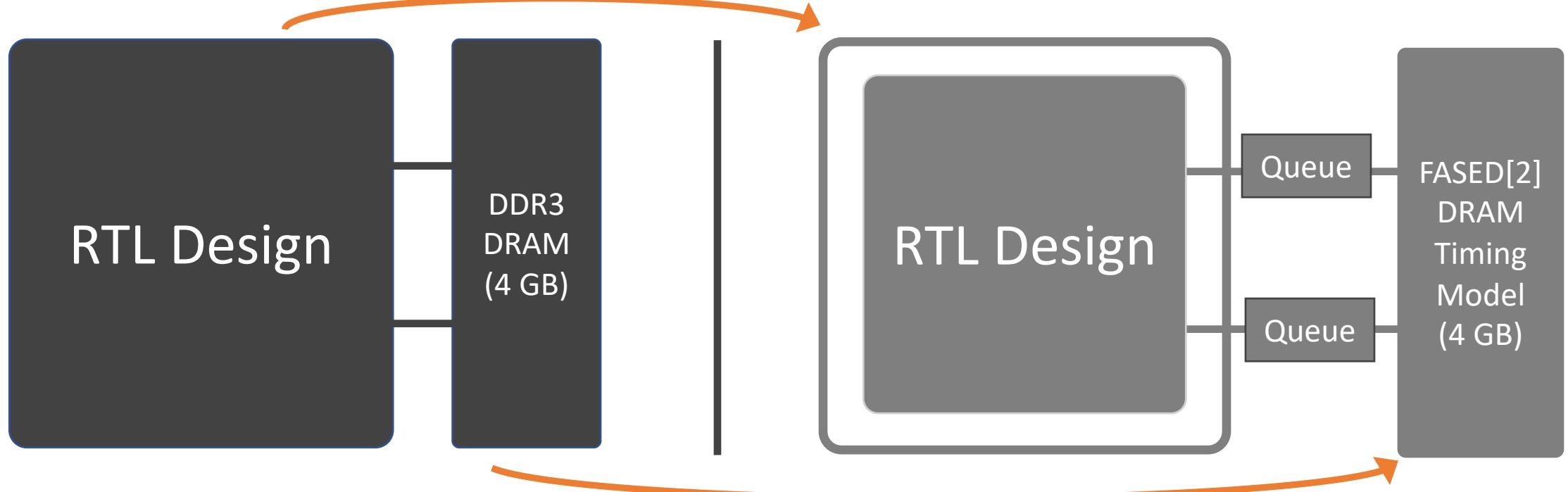
FireSim Generates FPGA-Hosted Simulators





Host Decoupling in FireSim: Transforming the Target

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



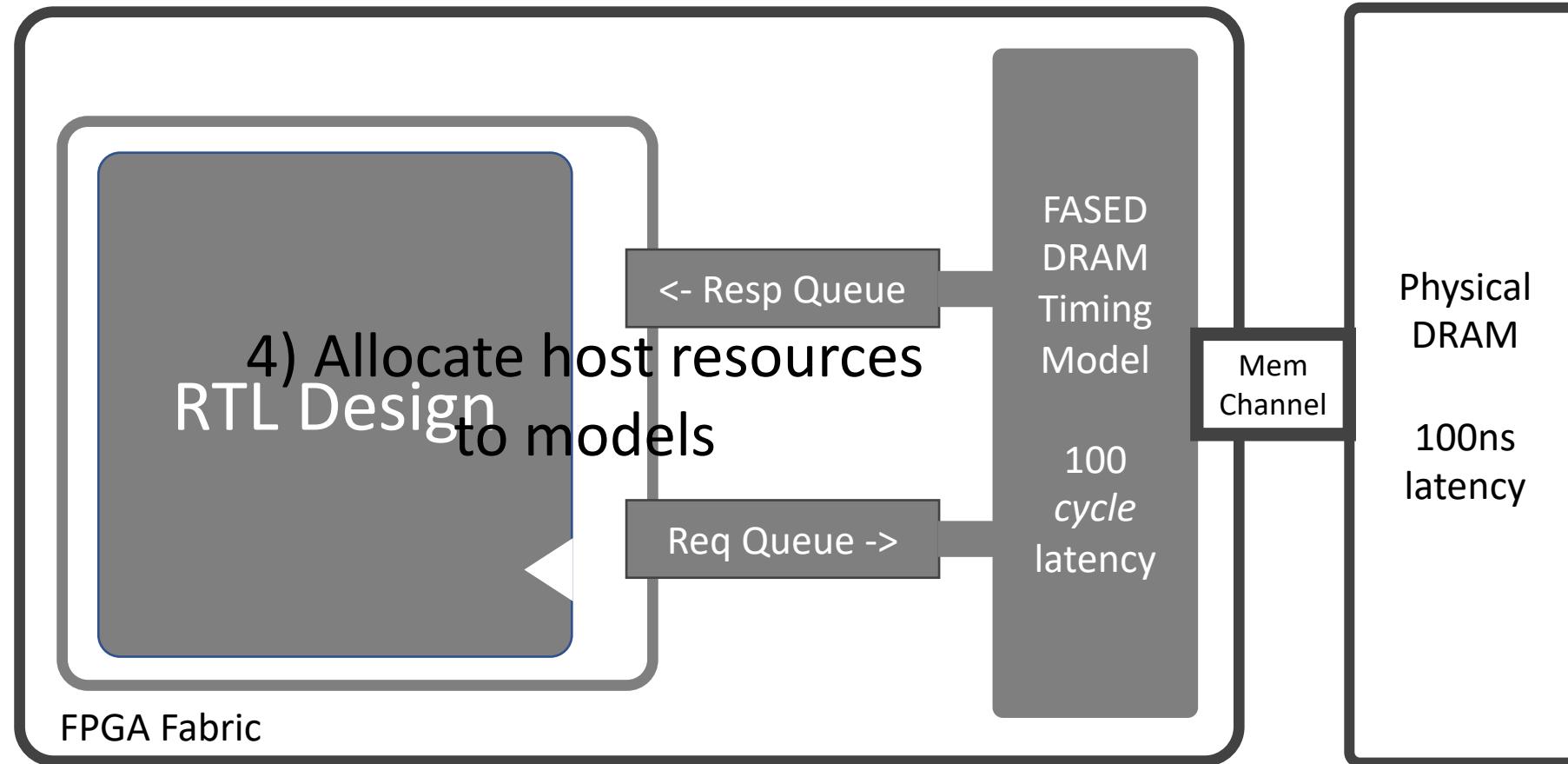
- 2) Generate FPGA-hosted model for DRAM[2] (think DRAMSim on an FPGA)
- 3) Generate queues (token channels) to connect the target models

[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





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 (ISCA '18)
3. Non-invasive debugging and instrumentation (FPL '18, ASPLOS '20)
4. Multi-cycle resource optimizations (ICCAD '19)





The Two Key Pieces of FireSim

1) The Compiler → Golden Gate

- Consumes RTL (FIRRTL) and generates emulator sources

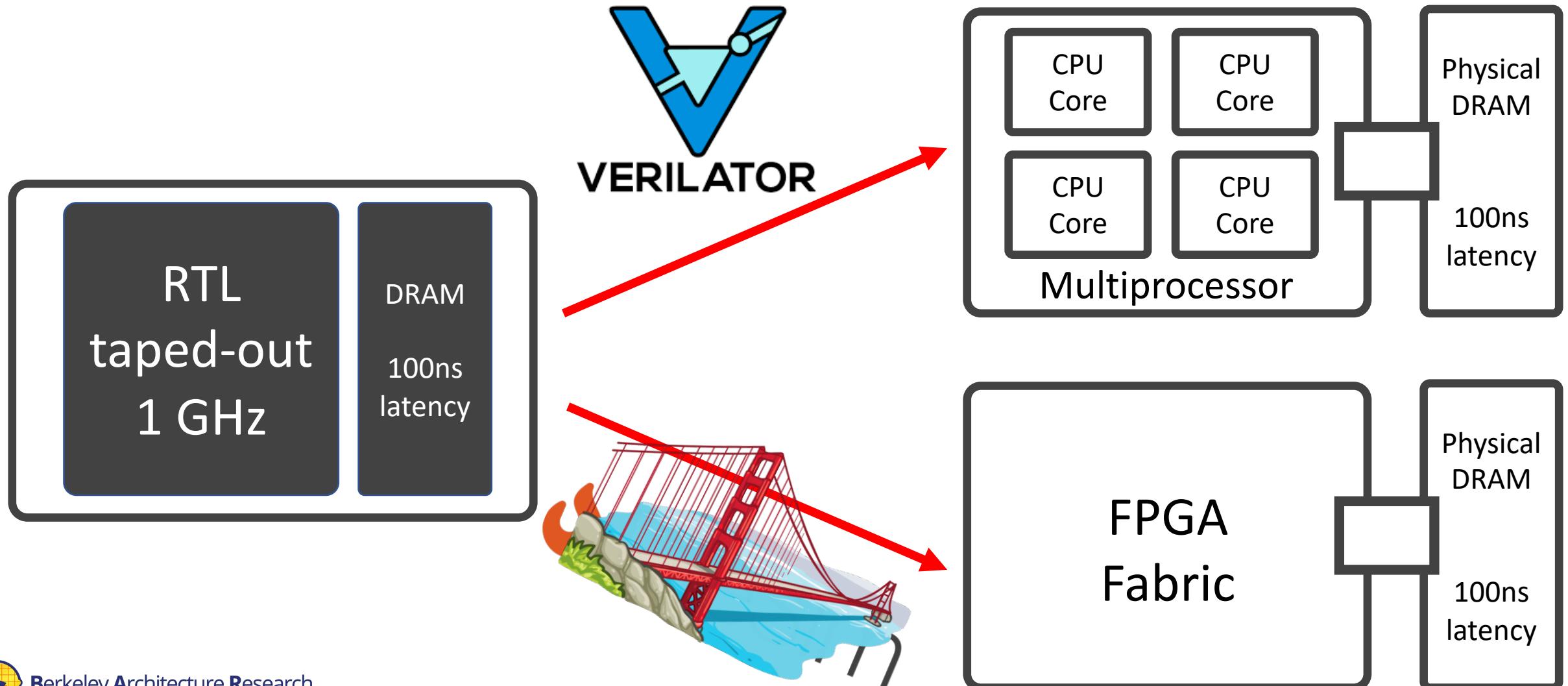
2) The Manager → firesim

- Puts everything together, distributes out to the cloud:
 - simulator builds (GG compilation, Bitstream generation)
 - simulations (workloads, FPGA images, SW drivers, etc.)





Golden Gate: The Analogy From Before





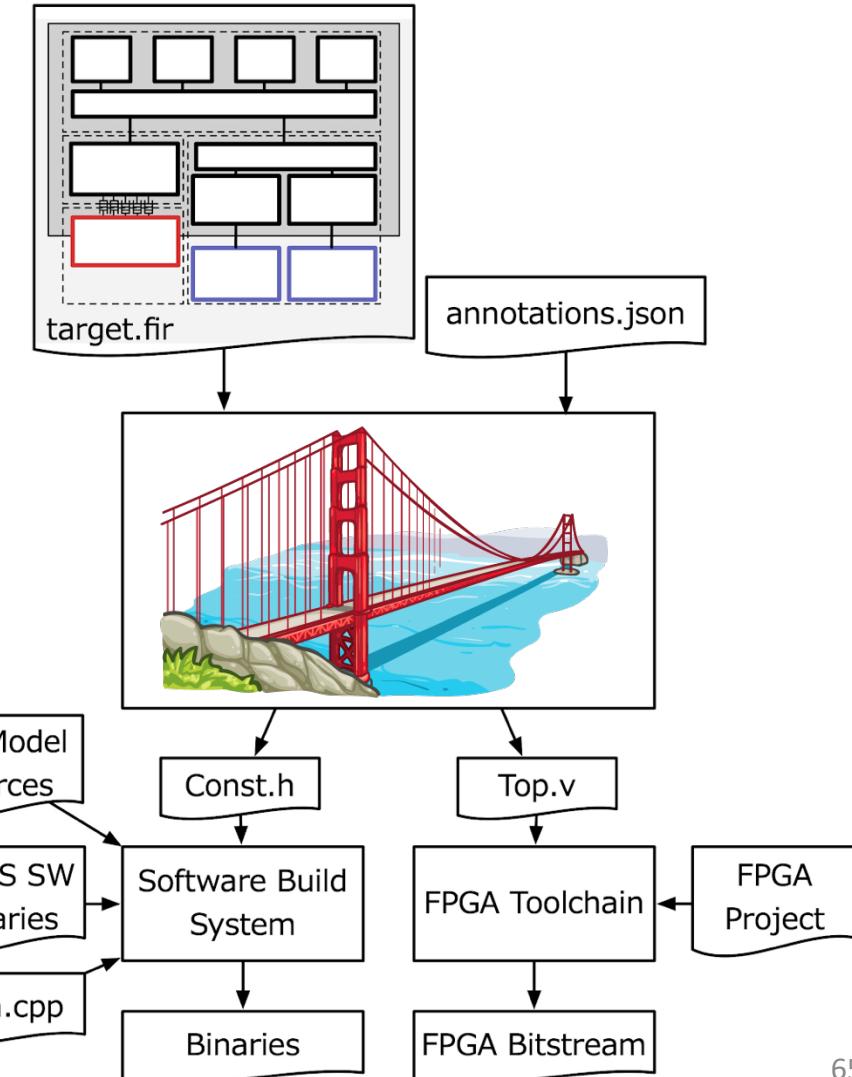
Golden Gate Compiler

Inputs:

- FIRRTL & annos from a Chipyard generator
- Compiler configuration

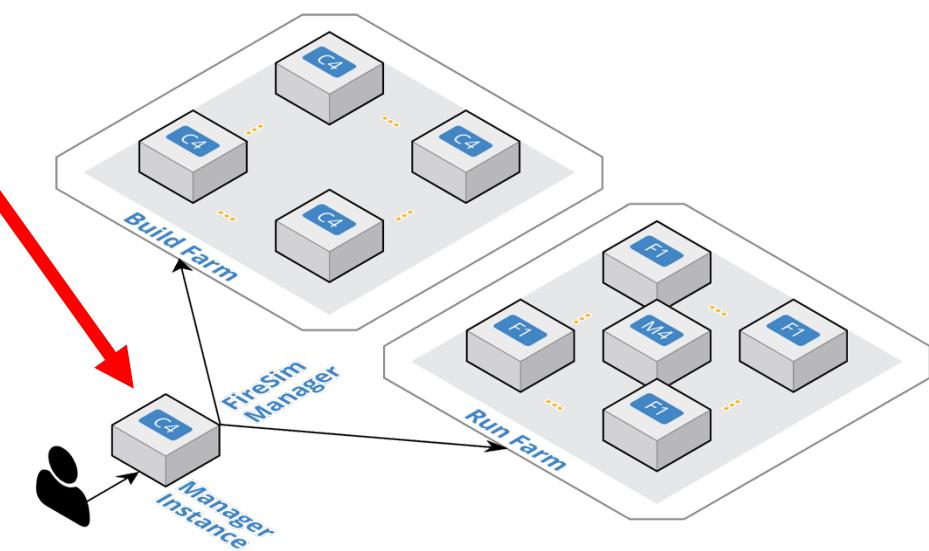
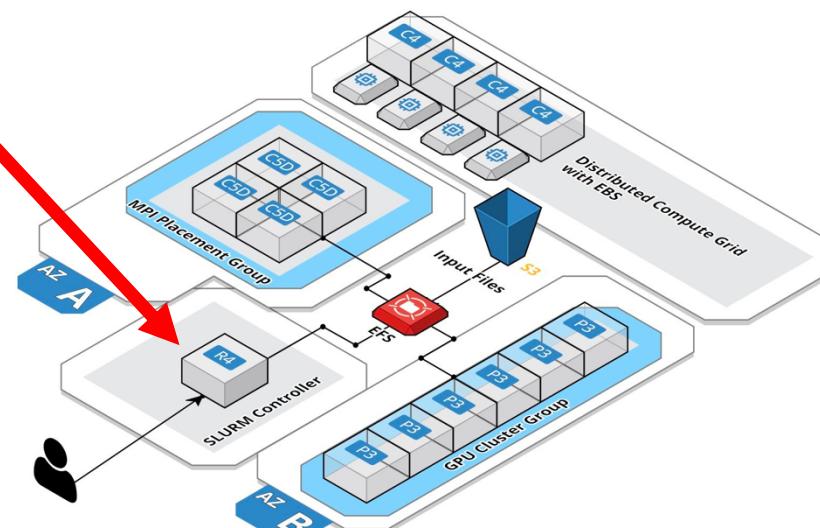
→ Produces sources for a simulator that are:

- deterministic
- support co-simulation of software models
- *area-optimized to fit more on the FPGA*



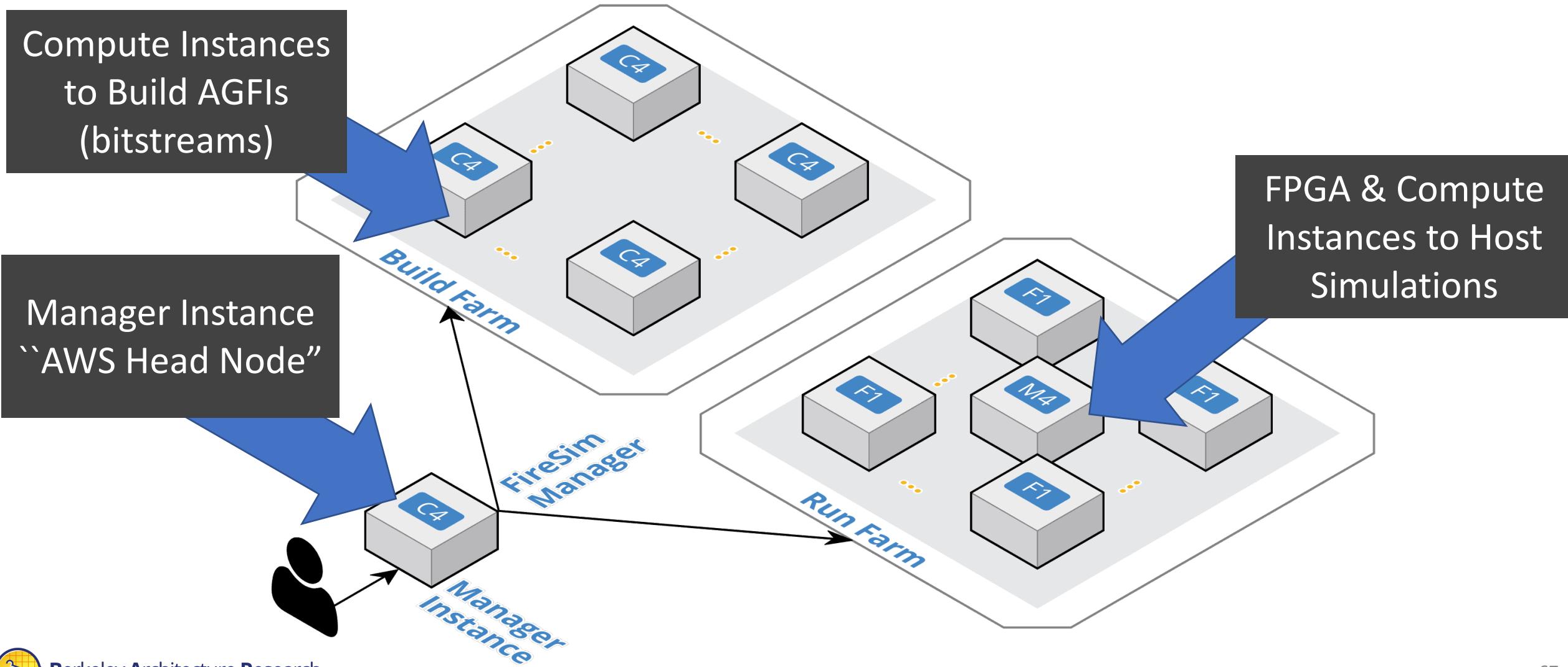


The Manager: A Second Analogy



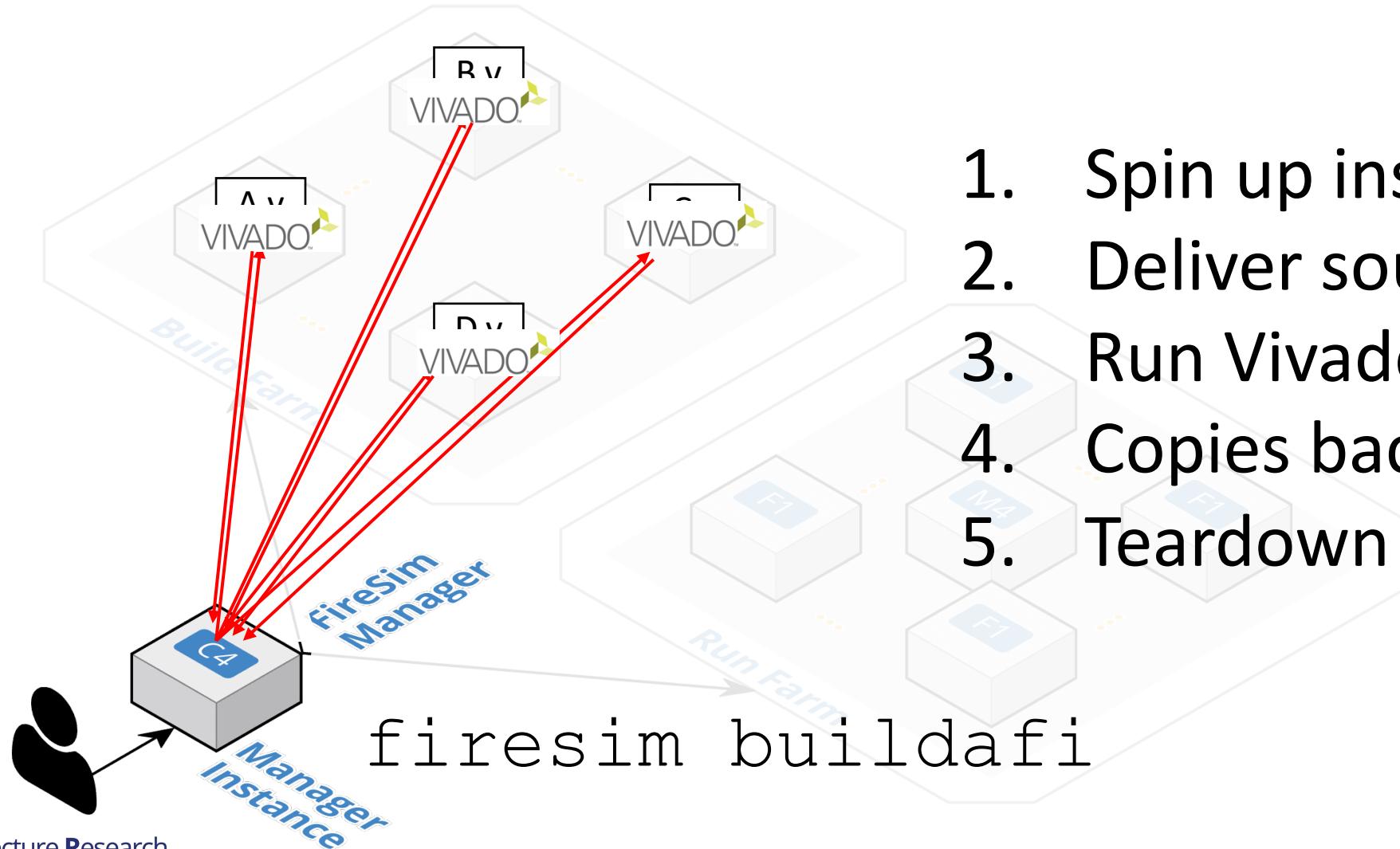


How Does FireSim use EC2?





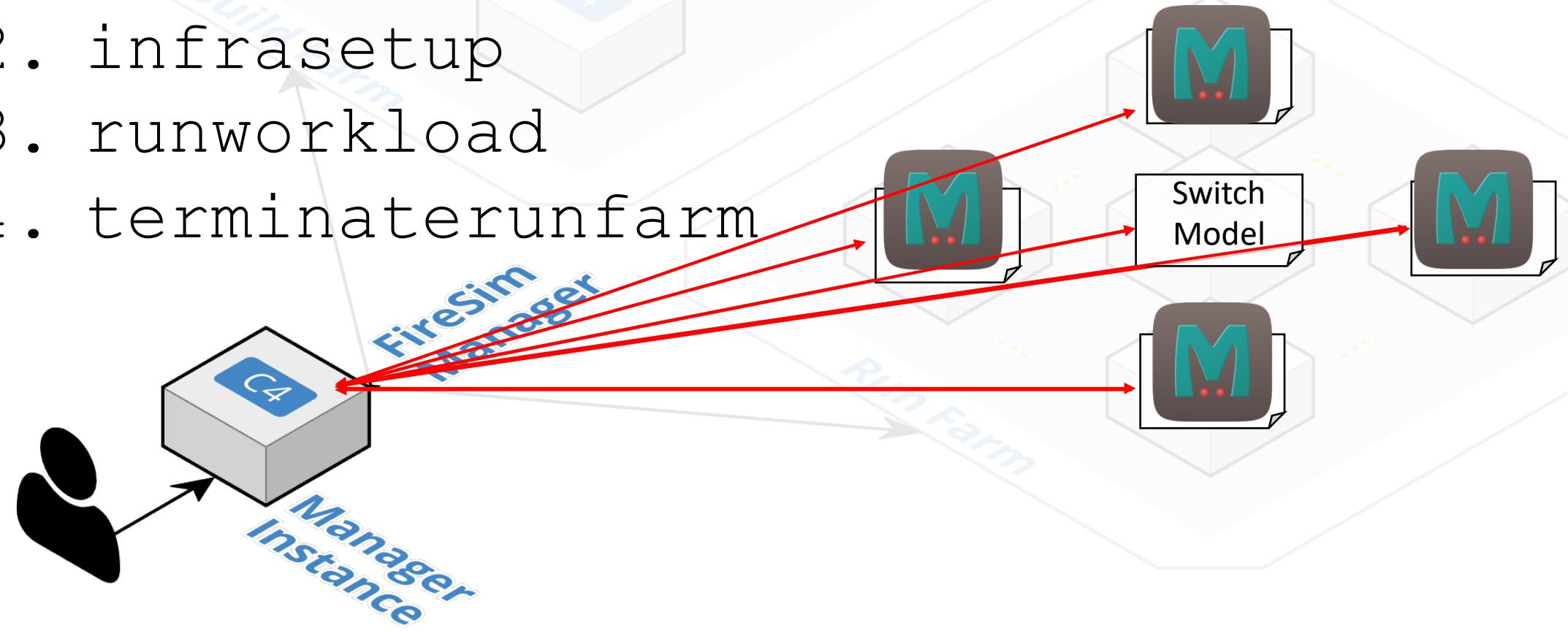
Automated Simulator Builds





Automated Simulation Management

1. launchrunfarm
2. infrasetup
3. runworkload
4. terminaterunfarm





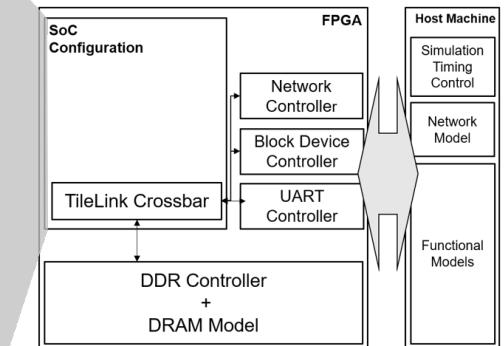
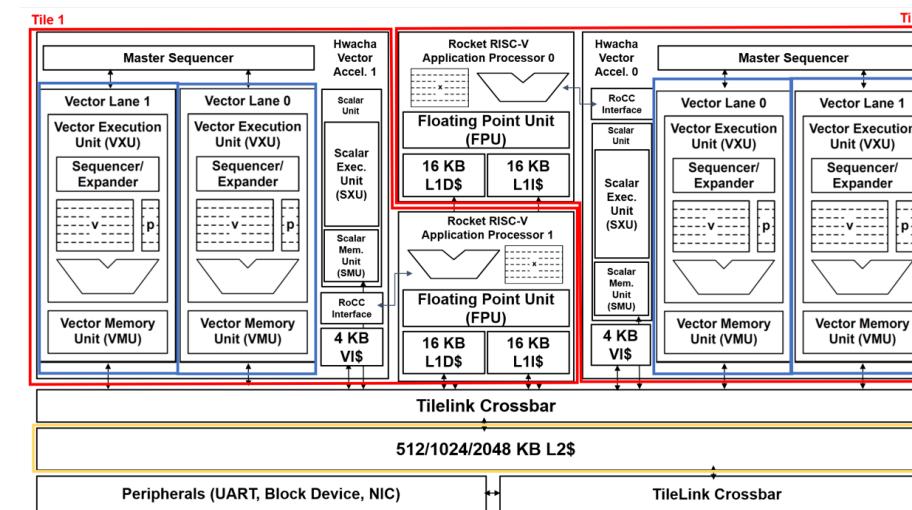
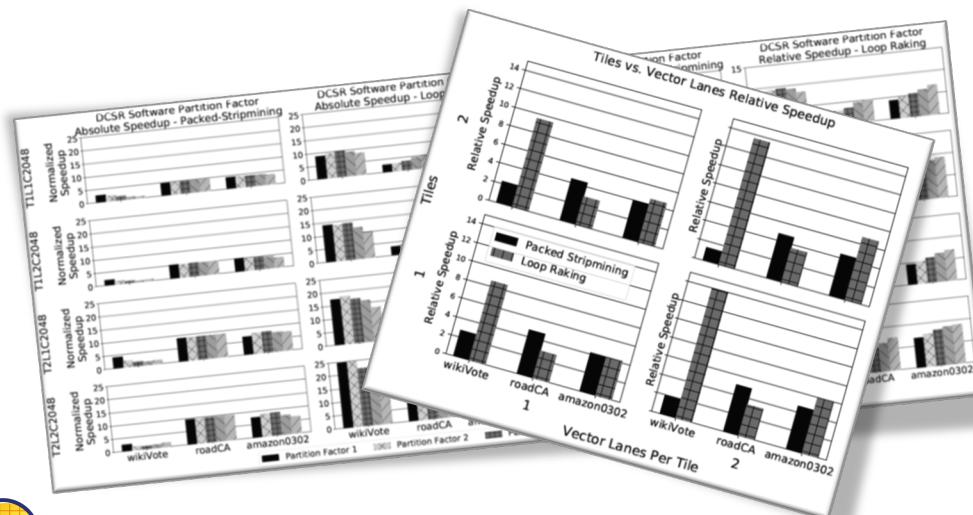
FireSim Features and Example use-cases





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

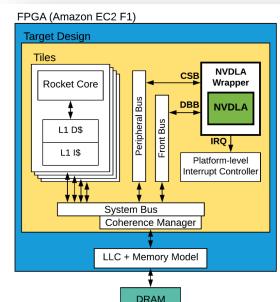
- Security
 - Replicate and identify uarch-level attacks, using pre-silicon RTL
 - BOOM Spectre replication
- Accelerator prototyping (integrated with Rocket Chip)
 - Chisel-based ML accelerators (Gemmini)
 - Open-source accelerator evaluation (NVDLA)
 - HLS-based rapid prototyping (Centrifuge)



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

Farzad Farshchi
University of Kansas
farshchi@ku.edu

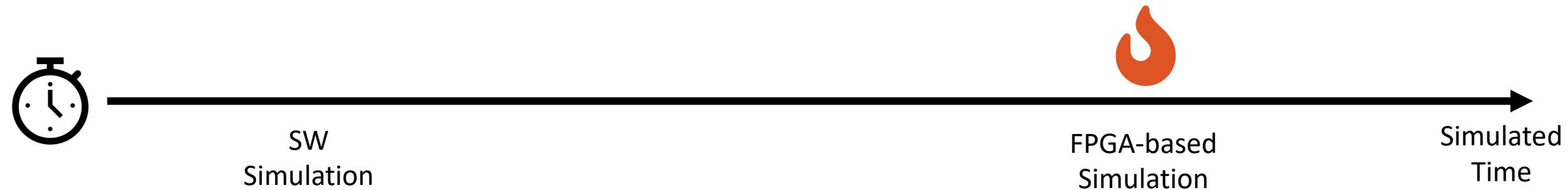
Qijing Huang
University of California,
berkeley
qijing.huang@berkeley.edu





Debugging and Profiling SoC Designs

- High simulation speed in FPGA-based simulation enables advanced debugging and profiling tools.
- Reach “deep” in simulation time, and obtain large levels of coverage and data



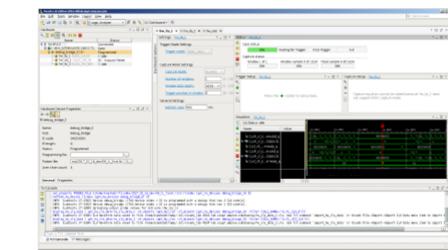


Debugging and Profiling SoC Designs

- AutoILA: 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 AutoILA 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.targetutils.FpgaDebug

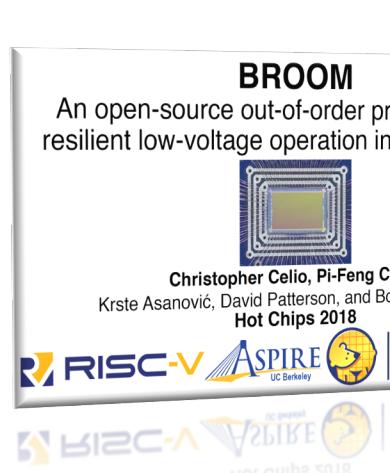
class SomeModuleIO(implicit p: Parameters) extends SomeIO()(p){
    val out1 = Output(Bool())
    val in1 = Input(Bool())
    FpgaDebug(out1, in1)
}
```





Debugging and Profiling SoC Designs

- Assertion Synthesis, Printf Synthesis
 - Common software debugging primitives
 - Automatic integration with simulation host
 - Assertions helped Identify BOOM bugs trillions of cycles into execution

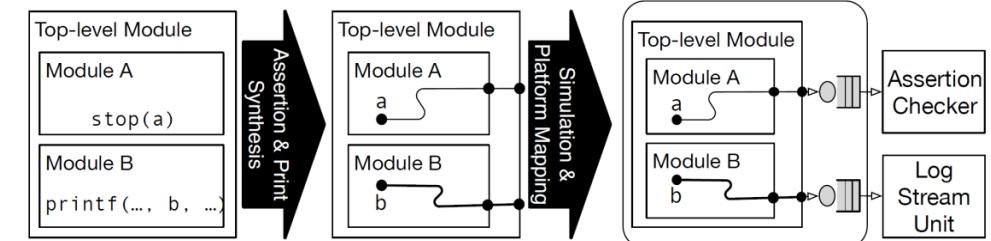


BOOM-v2 Assertion Results

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.omnnetpp.test	Pipeline hung	3.3	3.9
445.gobmk.test	Invalid write back in ROB	14.9	9.0
471.omnnetpp.ref	Pipeline hung	62.6	22.2
401.bzip2.ref	Wrong JAL target	473.7	164.6

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

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



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

Donggyu Kim¹, Christopher Celio², Sagar Karandikar¹, David Biancolin¹, Jonathan Bachrach¹, Krste Asanović¹

¹Department of Electrical Engineering and Computer Sciences, University of California, Berkeley

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

²Esperanto Technologies

christopher.celio@esperantotech.com

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





BOOM Example

- How it looks in the UART output (while Linux is booting):

```
[    0.008000] VFS: Mounted root (ext2 filesystem) on device 253:0.  
[    0.008000] devtmpfs: mounted  
[    0.008000] Freeing unused kernel memory: 148K  
[    0.008000] This architecture does not have kernel memory protection.  
mount: mounting sysfs on /sys failed: No such device  
Starting syslogd: OK  
Starting klogd: OK  
Starting mdev...  
mdev: /sys/dev: No such file or directory  
[id: 1840, module: Rob, path: FireBoom.boom_tile_1.core.rob]  
Assertion failed: [rob] writeback (0) occurred to an invalid ROB entry.  
    at rob.scala:504 assert (!(io.wb_resps(i).valid && MatchBank(GetBankIdx(rob_idx)) ) &&  
at cycle: 1112250469  
  
*** FAILED *** (code = 1841) after 1112250485 cycles  
time elapsed: 307.8 s, simulation speed = 3.61 MHz  
FPGA-Cycles-to-Model-Cycles Ratio (FMR): 2.77  
Beats available: 2165  
Runs 1112250485 cycles  
[FAIL] FireBoom Test  
SEED: 1569631756  
at cycle 4294967295
```

It would take ~62 hours to hit
this assertion is SW RTL
simulation (at 5 KHz sim rate),
vs. just a few minutes in FireSim



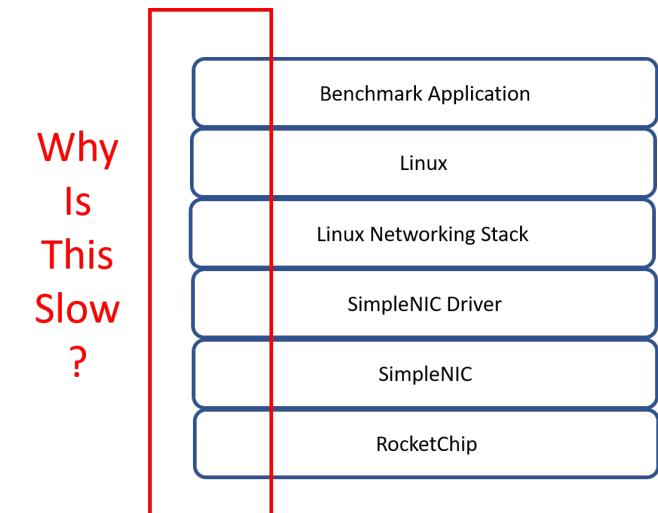
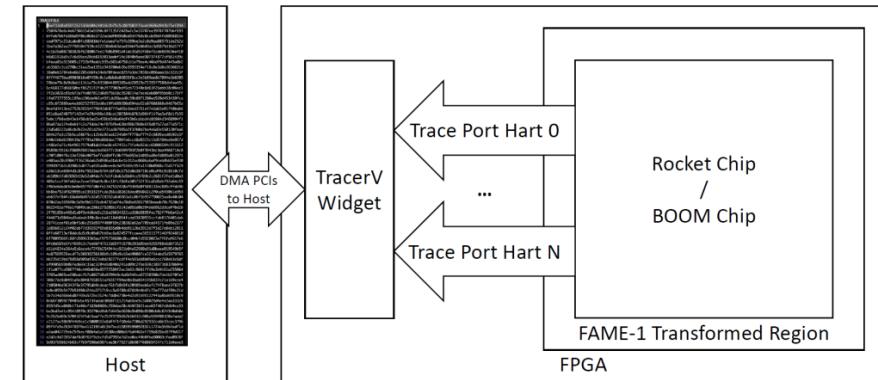


Debugging and Profiling SoC Designs

- TracerV: 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, Network Driver, Linux
 - Keystone Secure Enclave Project
 - High-performance hardware-specific code

More about this
in ASPLOS 2020:

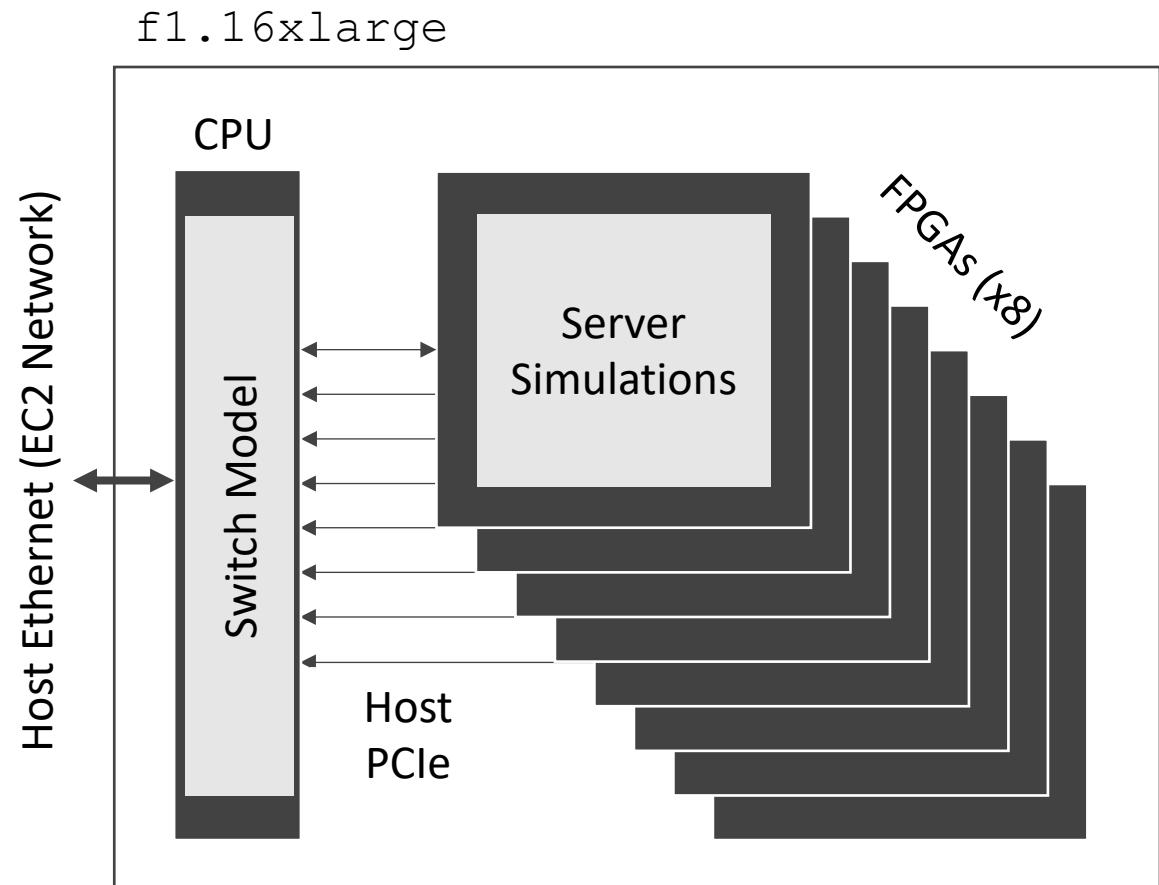
S. Karandikar, et al. “FirePerf: FPGA-Accelerated Full-System Hardware/Software Performance Profiling and Co-Design”. To appear in ASPLOS 2020.
<https://sagark.org/assets/pubs/fireperf-asplos2020.pdf>





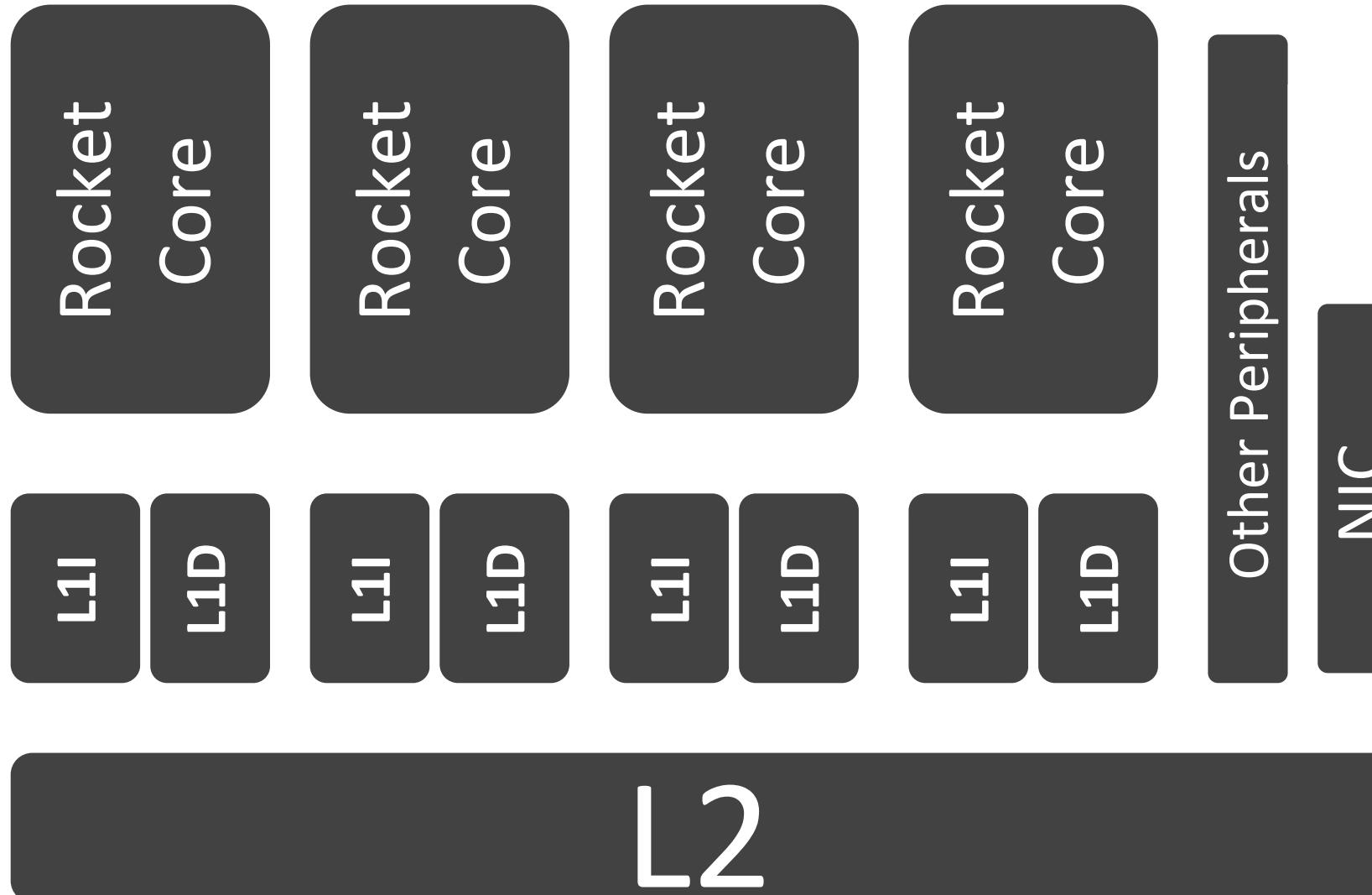
Scaling-Out SoC Designs

- Scale-out simulation
 - Model hardware at scale, cycle-accurately
 - Run real software
- RTL and abstract SW model co-simulation
- Server Simulations
 - Good fit for the FPGA
 - We have tapeout-proven RTL: FAME-1 transform w/Golden-Gate
- Network simulation
 - Little parallelism in switch models (e.g. a thread per port)
 - Need to coordinate all the distributed server simulations
 - So use CPUs + host network





Step 1: Server SoC in RTL



Modeled System

- 4x RISC-V Rocket Cores @ 3.2 GHz
- 16K I/D L1\$
- 256K Shared L2\$
- 200 Gb/s Eth. NIC

Resource Util.

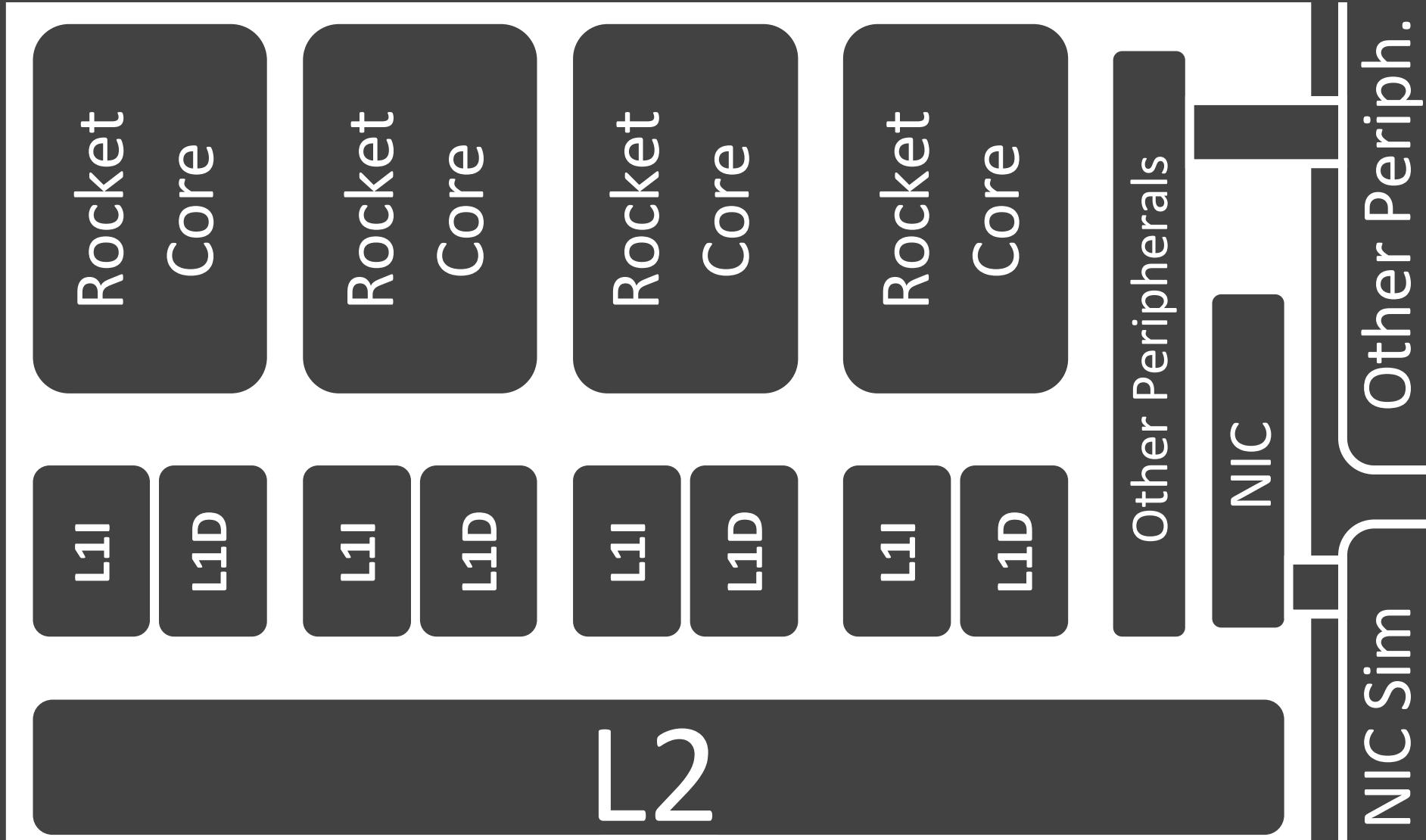
- < ¼ of an FPGA

Sim Rate

- N/A



Step 1: Server SoC in RTL



Modeled System

- 4x RISC-V Rocket Cores @ 3.2 GHz
- 16K I/D L1\$
- 256K Shared L2\$
- 200 Gb/s Eth. NIC

Resource Util.

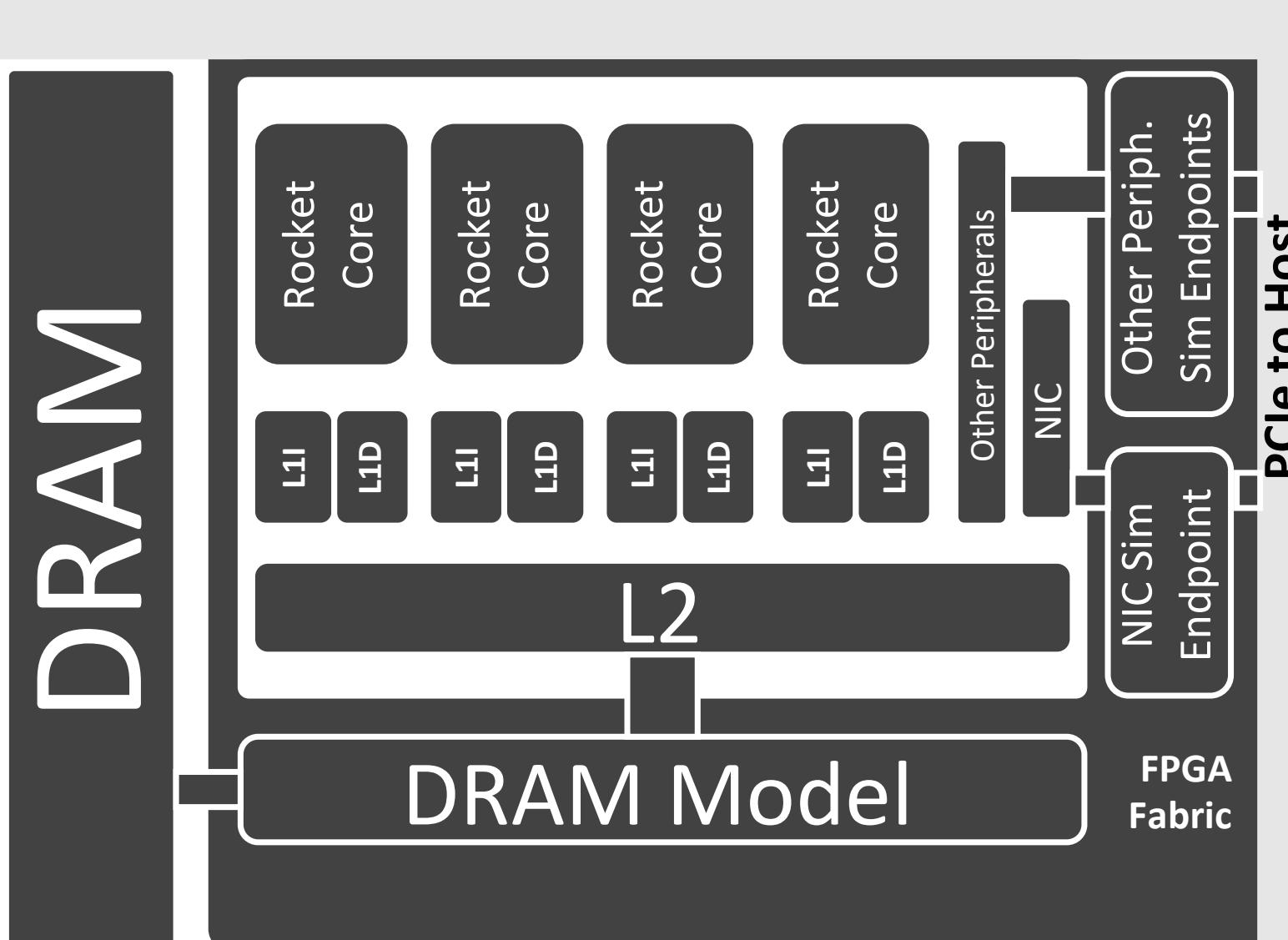
- < ¼ of an FPGA

Sim Rate

- N/A



Step 2: FPGA Simulation of one server blade



Modeled System

- 4x RISC-V Rocket Cores @ 3.2 GHz
- 16K I/D L1\$
- 256K Shared L2\$
- 200 Gb/s Eth. NIC
- 16 GB DDR3

Resource Util.

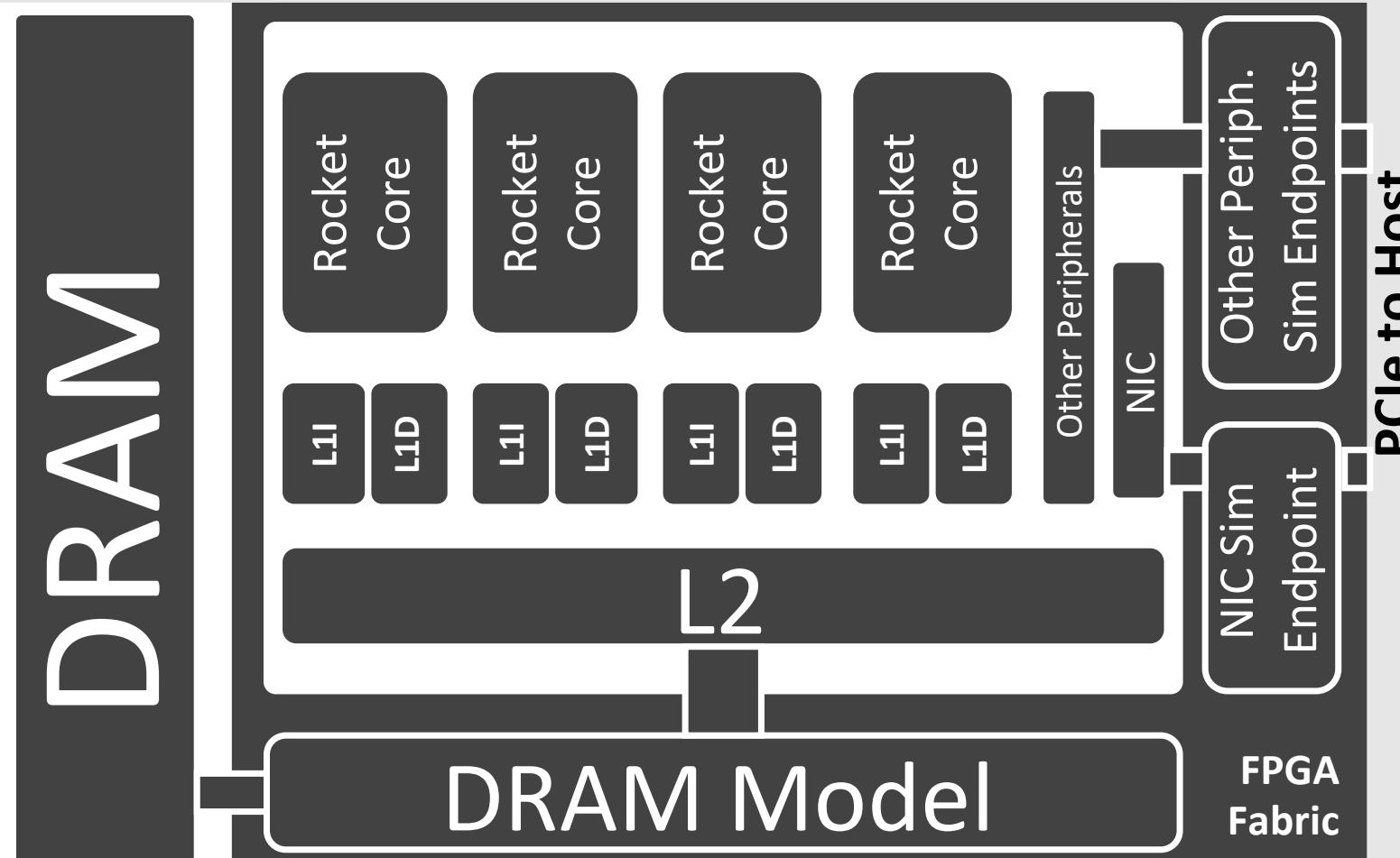
- < $\frac{1}{4}$ of an FPGA
- $\frac{1}{4}$ Mem Chans

Sim Rate

- ~150 MHz
- ~40 MHz (netw)



Step 2: FPGA Simulation of one server blade



Modeled System

- 4x RISC-V Rocket Cores @ 3.2 GHz
- 16K I/D L1\$
- 256K Shared L2\$
- 200 Gb/s Eth. NIC

Resource Util.

- < $\frac{1}{4}$ of an FPGA
- $\frac{1}{4}$ Mem Chans

Sim Rate

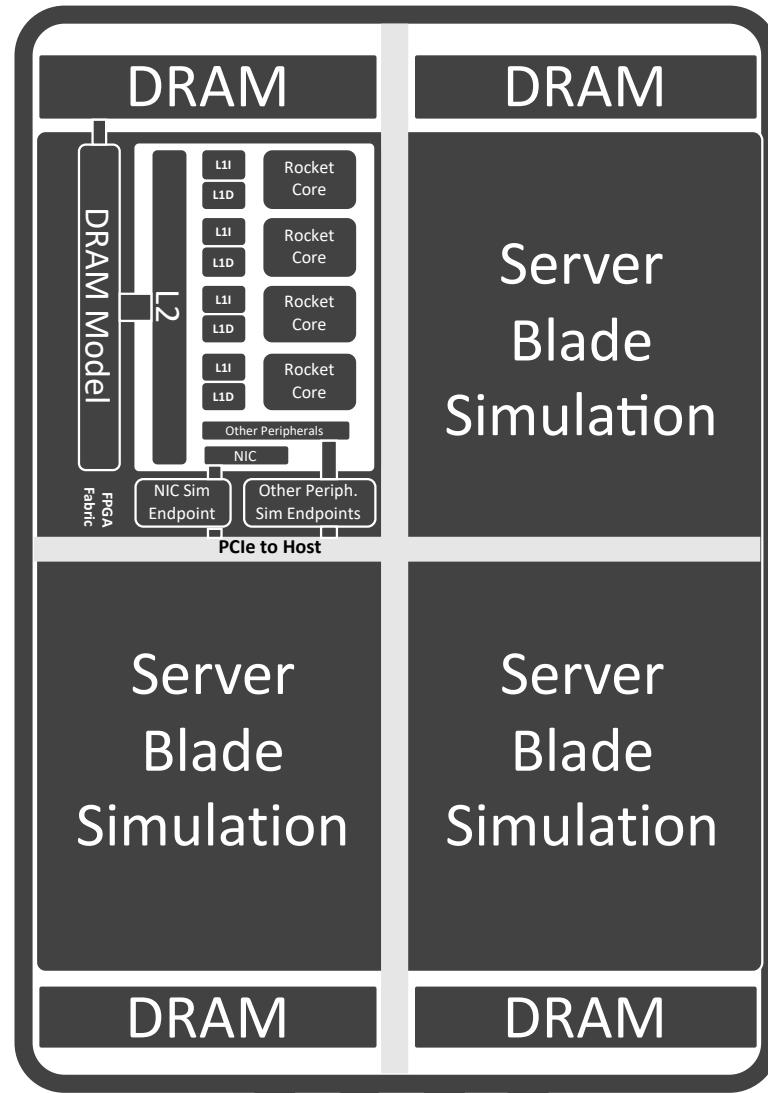
- ~150 MHz
- ~40 MHz (netw)



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

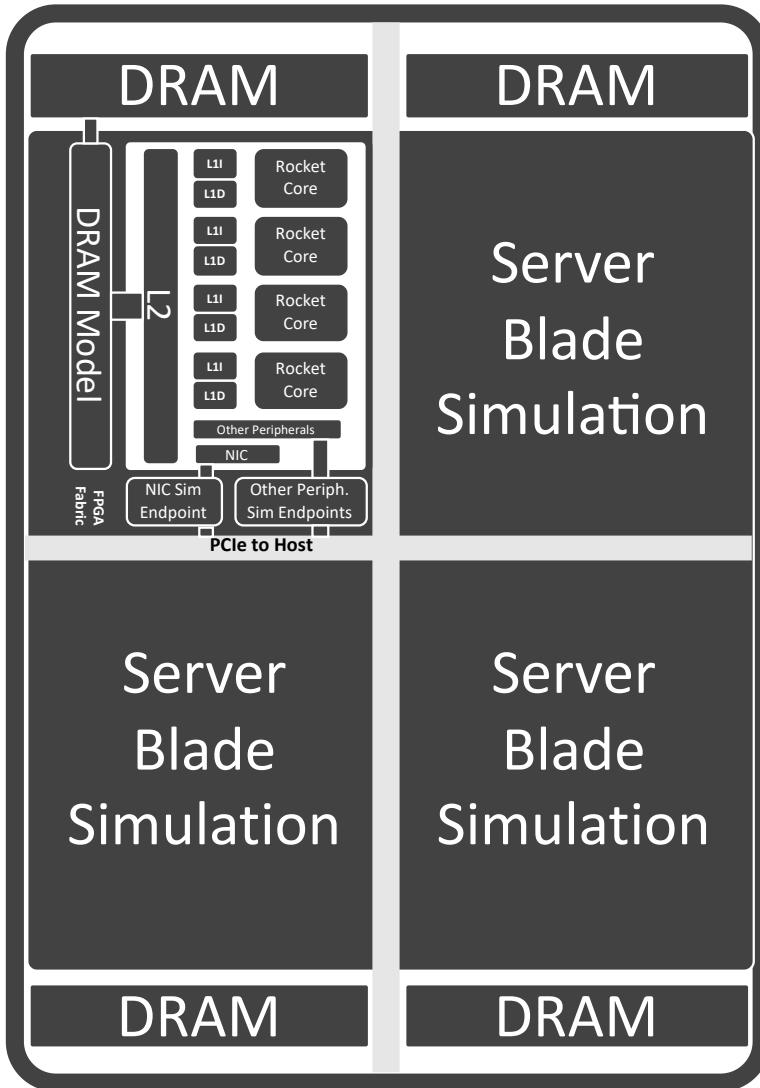
Sim Rate

- ~14.3 MHz
(netw)



Step 3: FPGA Simulation of 4 server blades

FPGA
4 Sims)



FPGA
(4 Sims)

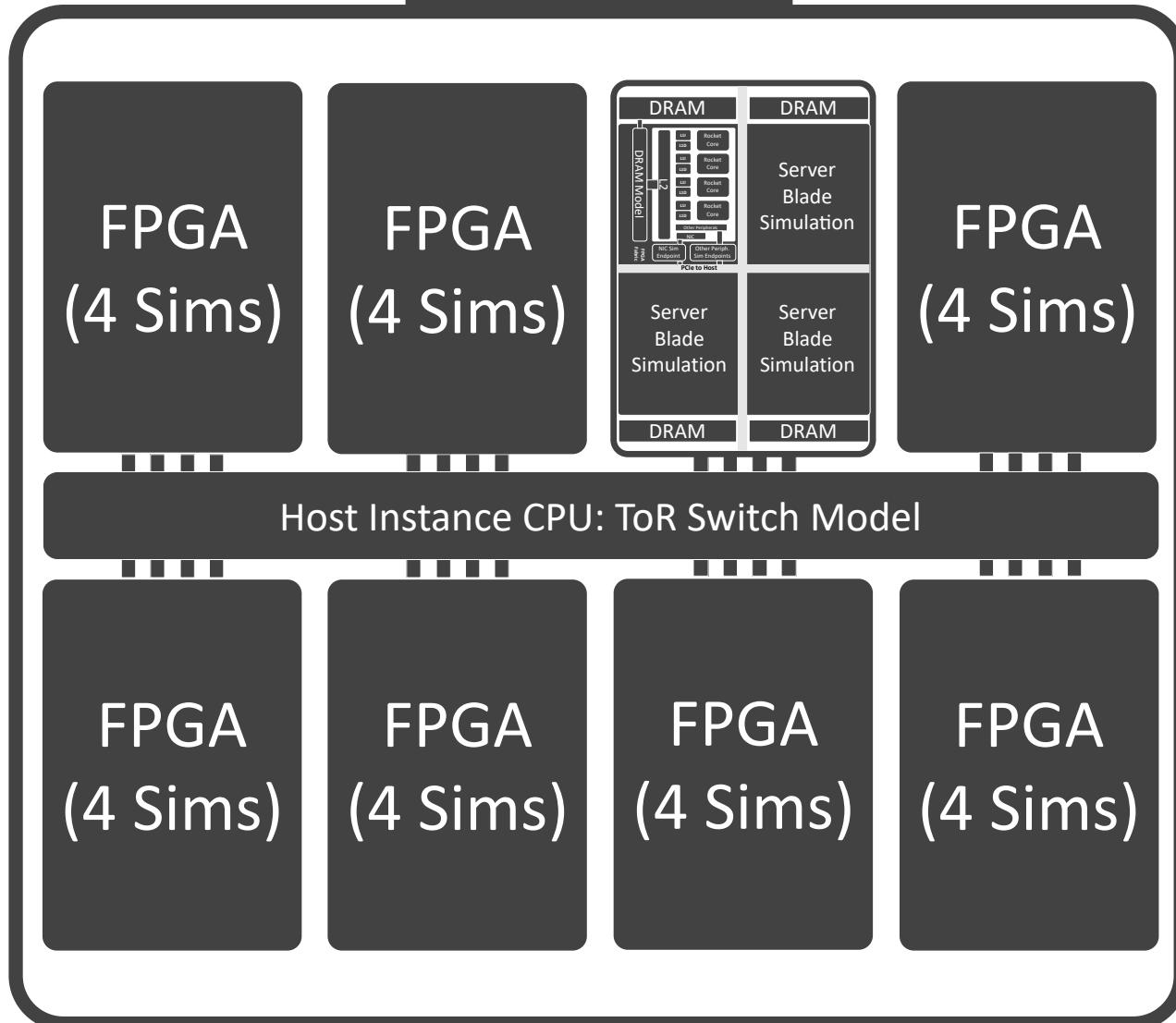
- Modeled System**
- 4 Server Blades
 - 16 Cores
 - 64 GB DDR3
- Resource Util.**
- < 1 FPGA
 - 4/4 Mem Chans
- Sim Rate**
- ~14.3 MHz (netw)



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

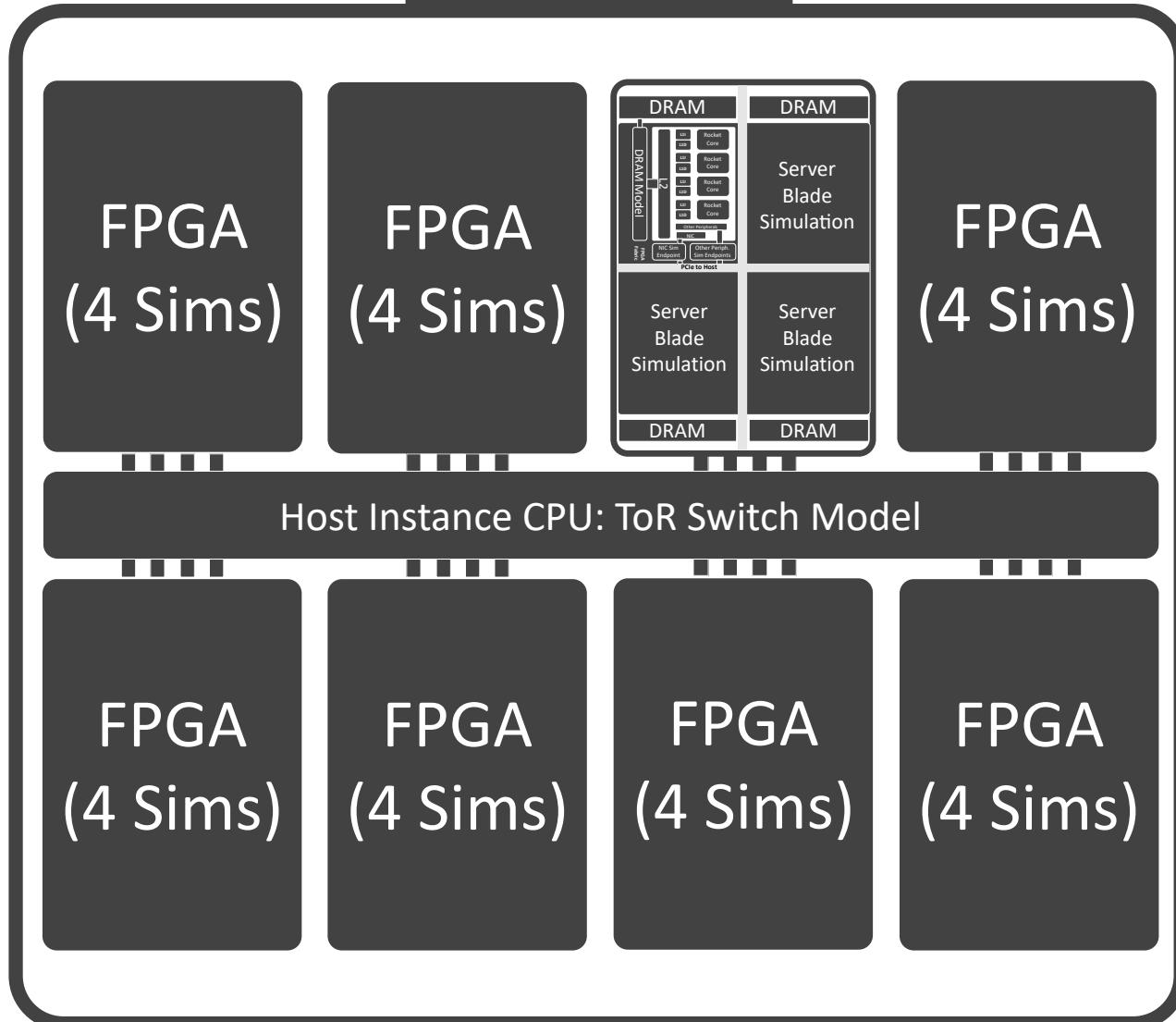
- ~10.7 MHz (netw)



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.

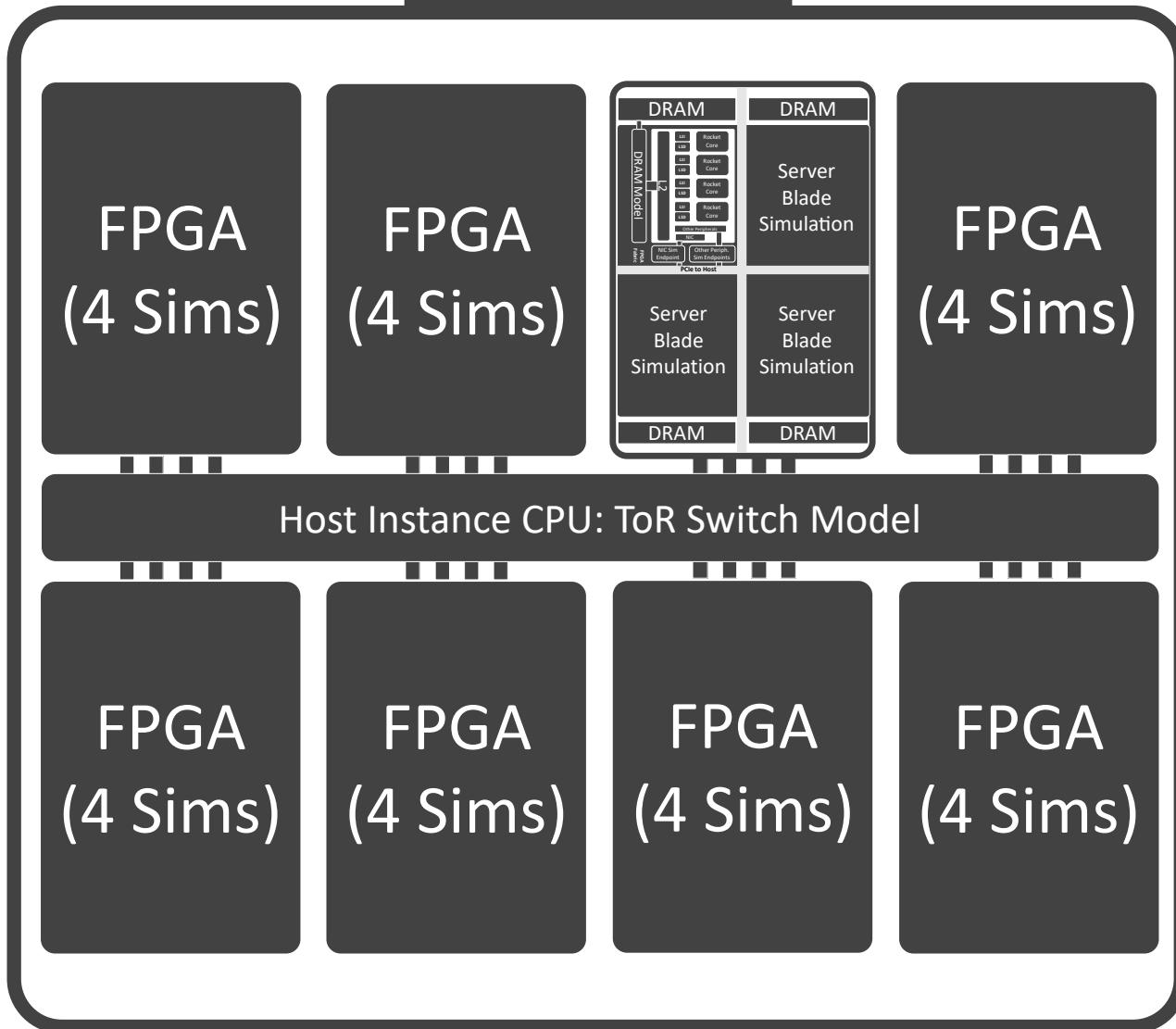
- 8 FPGAs =
- 1x f1.16xlarge

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.

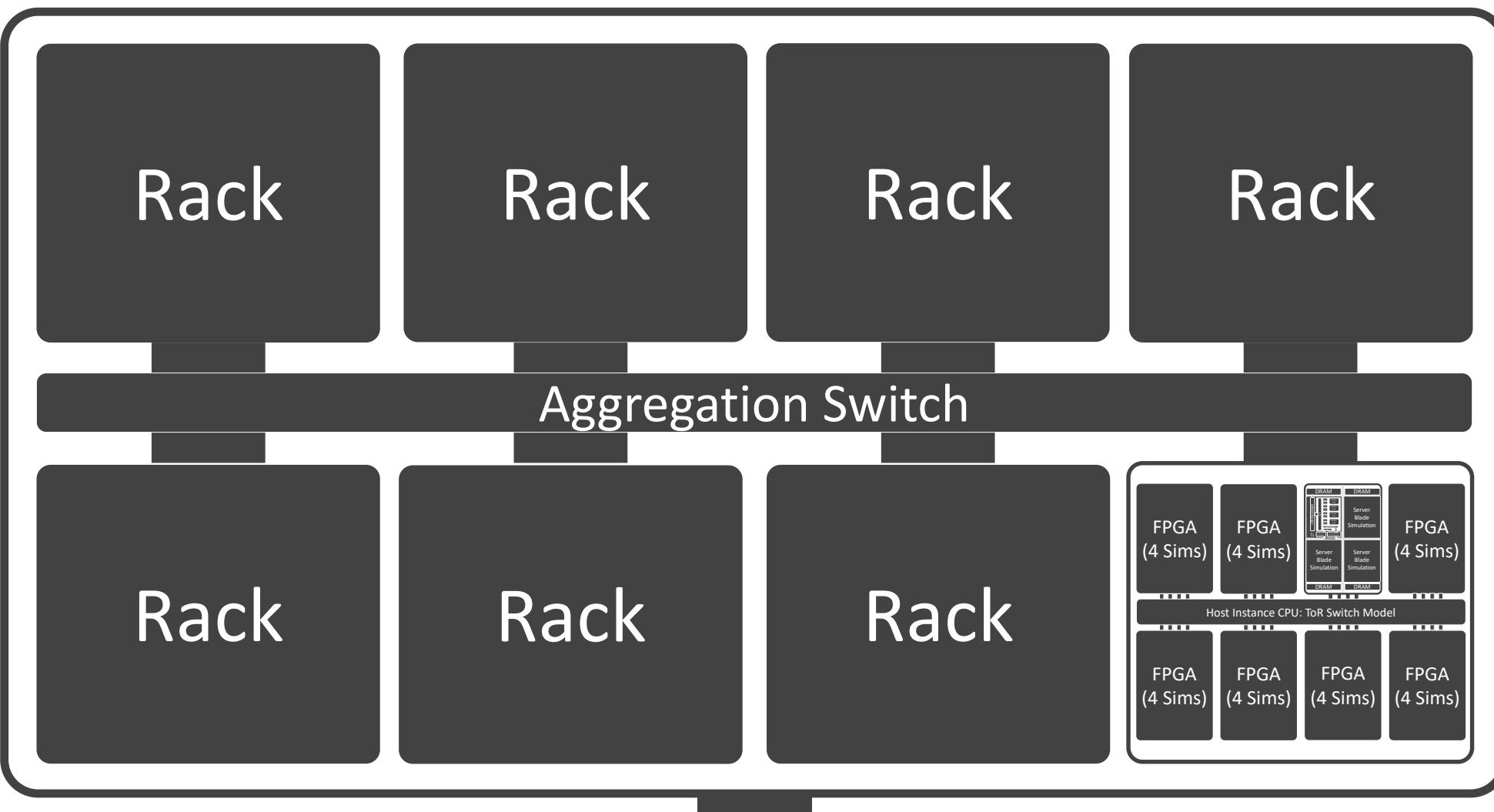
- 8 FPGAs =
- 1x f1.16xlarge

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.

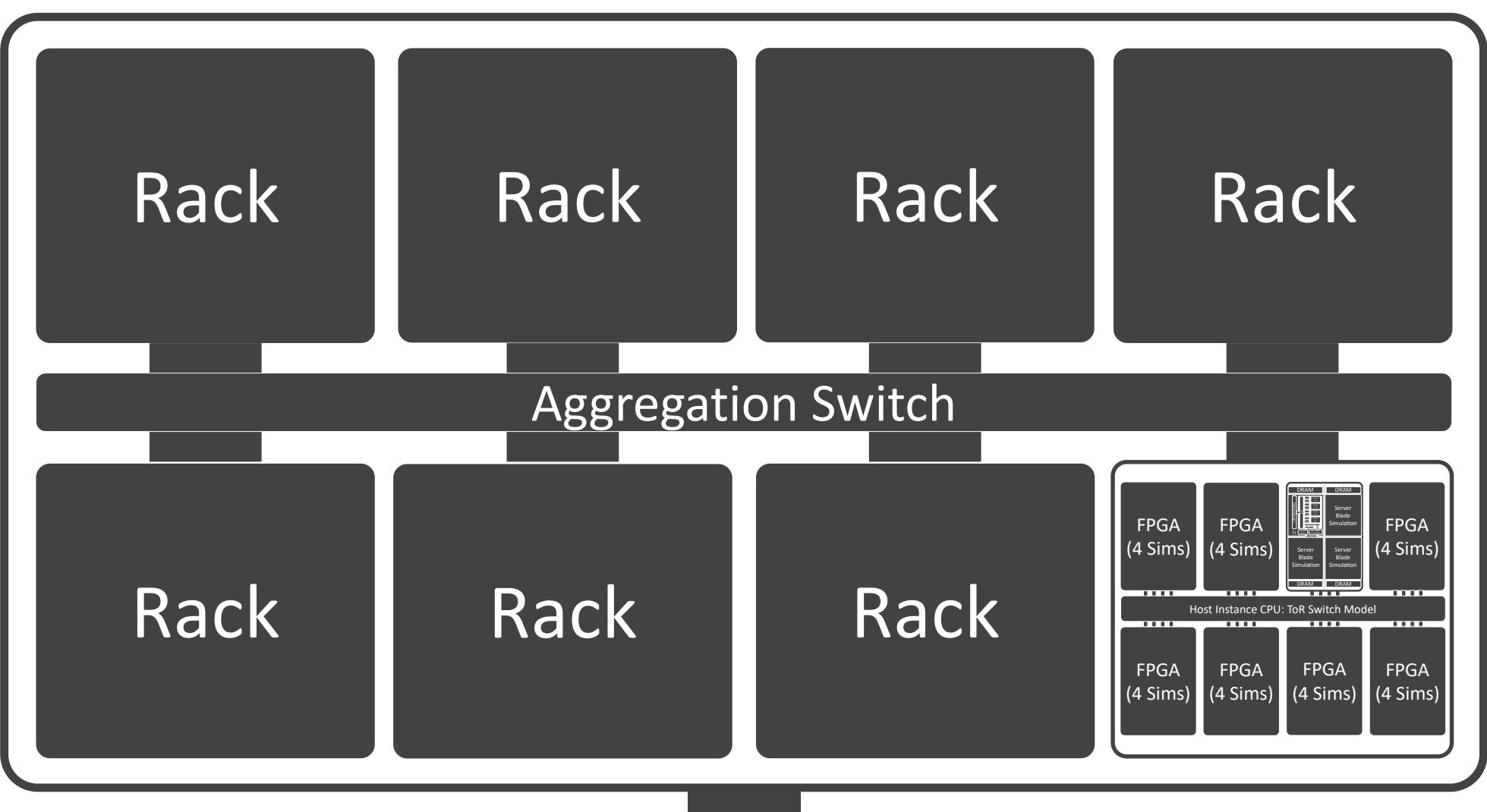
- 64 FPGAs =
- 8x f1.16xlarge
- 1x m4.16xlarge

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.

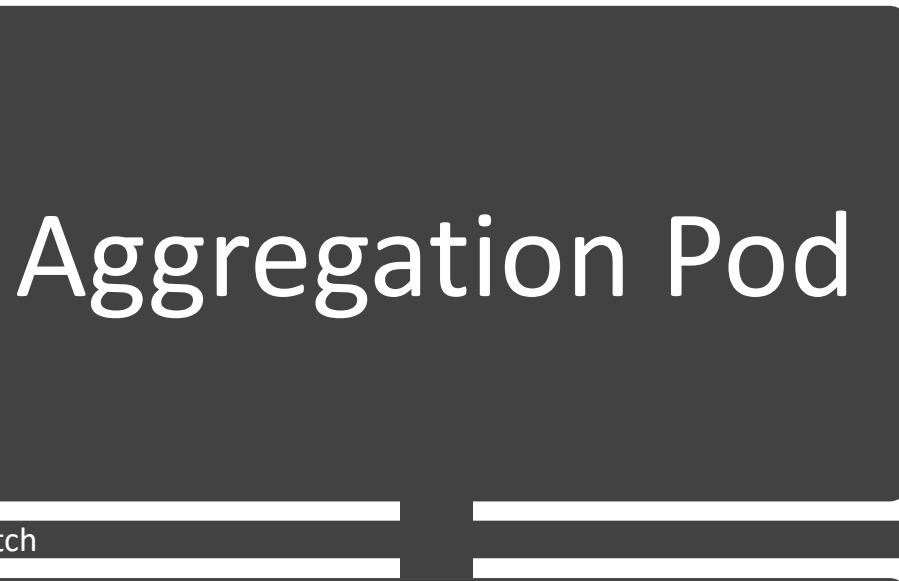
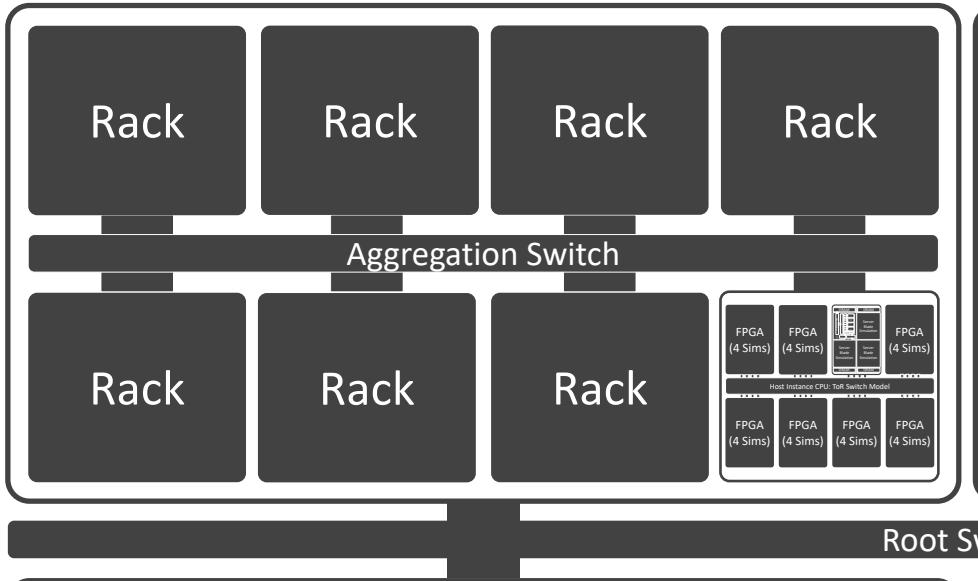
- 64 FPGAs =
- 8x f1.16xlarge
- 1x m4.16xlarge

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.

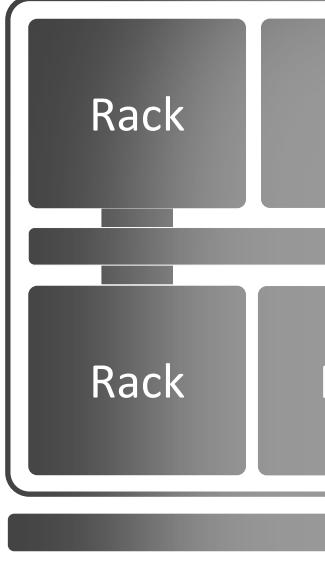
- 256 FPGAs =
- 32x f1.16xlarge
- 5x m4.16xlarge

Sim Rate

- ~6.6 MHz (netw)



Step 6: Simulating a 1024 node datacenter



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*

Aggregation Pod

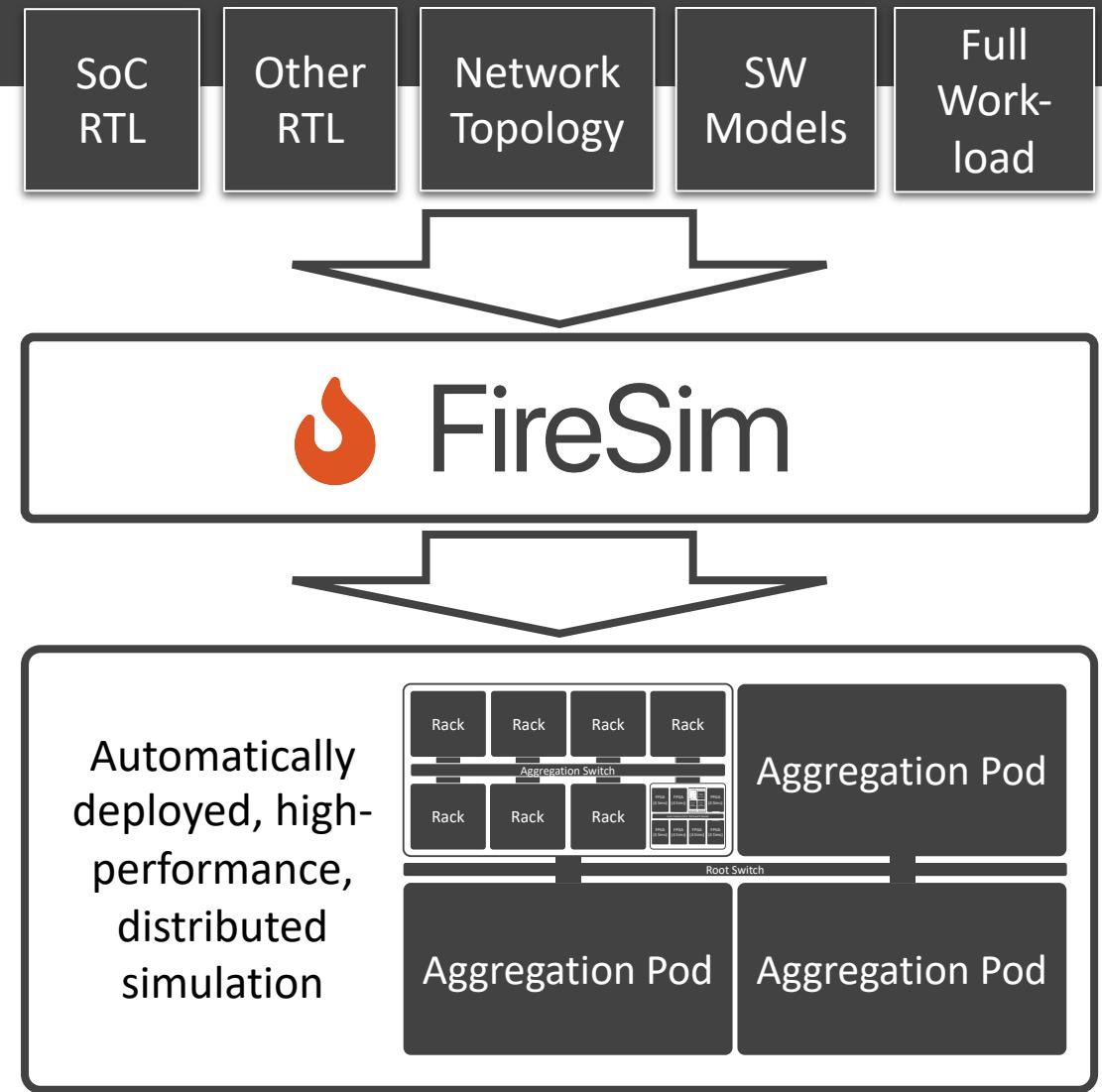
Aggregation Pod

Modeled System	
- 1024 Servers	
6 Cores	
1TB DDR3	
ToRs, 4 Aggr, 1	
Gb/s, 2us	
source Util.	
250 FPGAs =	
- 32x f1.16xlarge	
- 5x m4.16xlarge	
Sim Rate	
- ~6.6 MHz (netw)	

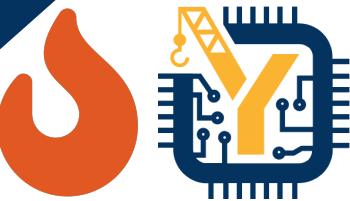


FireSim Recap

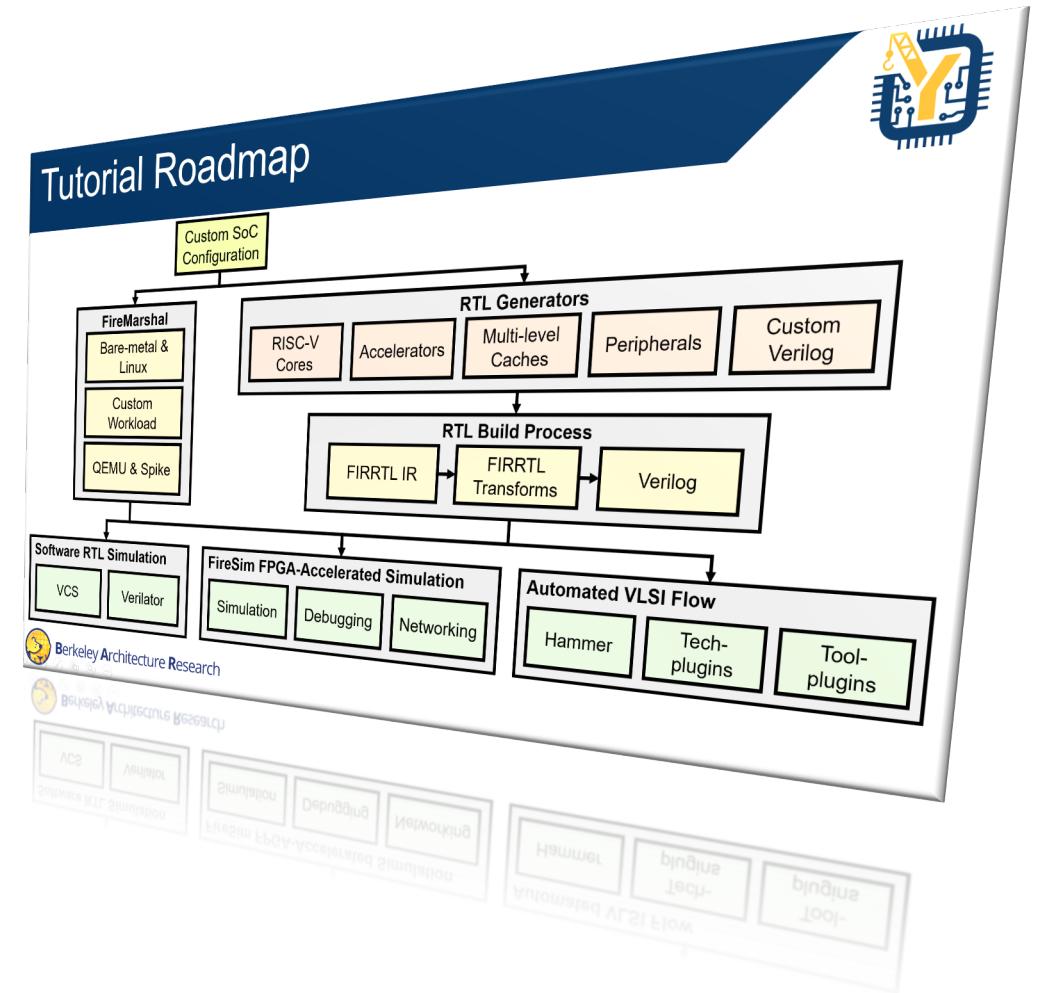
- We can prototype scalable-systems built on **arbitrary RTL** at **unprecedented scale**
 - + Mix software models when desired
- Simulation is **automatically built and deployed**
- Automatically **deploy real workloads** and collect results
- **Open-source**, runs on Amazon EC2 F1, **no capex**
- **Evaluate, Debug, Scale-out**



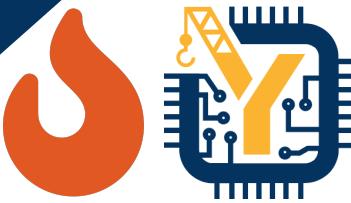
Recap



- End-to-End Evaluation
- Chipyard
 - Multi-flow infrastructure
 - Composing SoC using generators
 - Adding and simulating custom accelerators
 - Hammer VLSI flow
- FireSim
 - Full-system FPGA-accelerated simulation
 - Compiler and manager components
 - Debugging and instrumentation
 - Network simulation



Learn More



- Chipyard
 - GitHub: <https://github.com/ucb-bar/chipyard/>
 - Docs: <https://chipyard.readthedocs.io/en/latest/index.html>
 - Mailing List: <https://groups.google.com/forum/#!forum/chipyard>
- FireSim
 - Website: <https://fires.im/>
 - GitHub: <https://github.com/firesim/firesim/>
 - Docs: <https://docs.fires.im/en/latest/>
 - Mailing List: <https://groups.google.com/forum/#!forum/firesim>



Amazon Web Services (AWS)

FireSim

CHIPYARD



Berkeley
Architecture
Research



AWS and FireSim

- AWS benefits for FPGA-accelerated simulation
 - Pay for usage - No capital investment for large expensive FPGA
 - Elasticity - Enable to run many parallel simulations
 - Collaborative infrastructure – ya'll are going to use an FPGA image/bitstream that the course staff prepared – sharing is relatively easy
- In a research setting, users may have a large amount of AWS credits. In a class setting, we need to be more resource-conscious
- Initial FireSim AWS infrastructure setup
 - Takes a while, but it's a one-time cost
 - Setup procedure designed to minimize potential for mistakes
 - We'll now go through it step-by-step





Lab 3 Instructions – AWS Setup

Set-up an AWS account,
based on the instructions
in the matching page on
the FireSim docs

EE 290-2 Spring 2020

Lab 3: Tiling and Optimization for Accelerators

1 Introduction

This lab will provide you with hands-on experience on the implications of mapping large matrix multiplication operations onto 2D systolic array accelerators, and give you experience using the Amazon Web Services (AWS) EC2 public cloud for FPGA-accelerated simulation.

As most neural network models do not fit in on-chip memory, loop blocking/tiling is an important tool in writing a performant neural network implementation. The additional degree of freedom afforded by the scratchpad in ML accelerators requires further planning of data re-use within the scratchpad. Furthermore, the size of the compute array adds an additional constraint on the tiling hierarchy.

There is a wide body of literature on loop blocking and scheduling for standard scalar processors. However, this body of literature is less extensive with regards to on-chip accelerators with dedicated memories that may be connected to the memory hierarchy in various forms.

Specifically, the DMA of the Gemmini accelerator is currently connected to the shared L2 cache of the scalar host processor. As such, it may see caching affects from the tiling scheme, as well as other parts of the program.

In this lab you will continue using the Chipyard and Gemmini platforms from the previous lab to further improve the performance of DNN execution using ML accelerators through software optimization.

1.1 FireSim and Amazon Web Services (AWS)

In order to use FireSim on AWS (you will find additional information about FireSim in the next section), you will need to open an AWS account. AWS is a paid service, so be careful about your usage (more details to follow). Do not wait with this process until the last minute, because it might have around a 1-day latency due to humans-in-the-loop. The lecture on Wednesday, February 26th, will provide a step-by-step tutorial describing the procedures in this section. We recommend following the instructions in [this page¹](#) of the FireSim documentation.

you will receive an email from us with a \$200 AWS credit promo code. This amount should be more than sufficient for the tasks in this assignment, assuming you manage your resources and track your AWS expenses. In addition, you can receive another promo code of \$100 by signing up as a student in [AWS Educate²](#) with your [berkeley.edu](#) email address. Do not sign up for the AWS Educate Starter Kit, because you will not have access to the FPGA-based F1 instances which are required for this lab. You will need to redeem your AWS promo code credits by following the [instructions³](#).

Once you have opened an AWS account and applied your promo code for credits, follow the initial FireSim AWS infrastructure setup instructions in [this page⁴](#) of the FireSim documentation. Do not continue to setting up your manager instance (we will provide you with a prepared manager AMI to shorten some of the build preparation process).

2 Background

2.1 Loop Tiling and Scheduling

We will be working on tiling matrix multiplication, since Gemmini accelerates matrix multiplication operations, and we saw in the previous lab that convolutions can be lowered to matrix multiplication operations. As a reminder, a standard nested-loop matrix multiplication operation looks as follows:

¹<https://docs.firesim/en/latest/Initial-Setup/First-time-AWS-User-Setup.html>

²<https://forms.gle/YLJKrgLGpdzpupk59>

³<https://aws.amazon.com/education/awseducate/>

⁴<https://aws.amazon.com/awscredits/>

⁵<https://docs.firesim/en/latest/Initial-Setup/Configuring-Required-Infrastructure-in-Your-AWS-Account.html>



FireSim Docs – AWS User Setup



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

First-time AWS User Setup

FireSim
latest

Search docs

GETTING STARTED:

- 1. FireSim Basics
- 2. Initial Setup/Installation
 - 2.1. First-time AWS User Setup
 - 2.1.1. Creating an AWS Account
 - 2.1.2. AWS Credit at Berkeley
 - 2.1.3. Requesting Limit Increases
 - 2.2. Configuring Required Infrastructure in Your AWS Account
 - 2.3. Setting up your Manager Instance
- 3. Running FireSim Simulations
- 4. Building Your Own Hardware Designs (FireSim FPGA Images)

ADVANCED DOCS:

- Manager Usage (the `firesim` command)
- Workloads
- Targets
- Debugging
- Supernode - Multiple Simulated SoCs Per FPGA
- Miscellaneous Tips
- FireSim Asked Questions

GOLDEN GATE (MIDAS II) DOCS:

Read the Docs v: latest ▾

Docs » 2. Initial Setup/Installation » 2.1. First-time AWS User Setup [Edit on GitHub](#)

2.1. First-time AWS User Setup

If you've never used AWS before and don't have an account, follow the instructions below to get started.

2.1.1. Creating an AWS Account

First, you'll need an AWS account. Create one by going to <aws.amazon.com> and clicking "Sign Up." You'll want to create a personal account. You will have to give it a credit card number.

2.1.2. AWS Credit at Berkeley

If you're an internal user at Berkeley and affiliated with UCB-BAR or the RISE Lab, see the [RISE Lab Wiki](#) for instructions on getting access to the AWS credit pool. Otherwise, continue with the following section.

2.1.3. Requesting Limit Increases

In our experience, new AWS accounts do not have access to EC2 F1 instances by default. In order to get access, you should file a limit increase request. You can learn more about EC2 instance limits here: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html#ec2-on-demand-instances-limits>

To request a limit increase, follow these steps:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html>

You'll probably want to start out with the following request, depending on your existing limits:

Limit Type: EC2 Instances



Berkeley Architecture Research

Opening an AWS Account



Create an AWS Account

The screenshot shows the official AWS website homepage. At the top right, there is a prominent orange button labeled "Create an AWS Account". This button is highlighted with a thick red rectangular border. The rest of the page includes the AWS logo, navigation links like "Products", "Solutions", "Pricing", etc., and a search bar. Below the header, there's a main banner for "Open Distro for Elasticsearch" with a subtext about searching logs. To the right of the banner is a graphic showing a magnifying glass over a document, surrounded by icons of people and a server rack. Below the banner are four smaller cards: "Amazon Lightsail" (a robot icon), "Amazon DynamoDB" (a database icon with a rocket), "Serverless Application Development" (a rocket launching icon), and "200,000+ Databases Migrated to AWS" (a database icon with the number 200,000).

Amazon Lightsail
Everything you need to get started on AWS—for a low, predictable price

Amazon DynamoDB
Fully managed nonrelational database for any scale

Serverless Application Development
Find tools for testing, deploying, and monitoring serverless applications

200,000+ Databases Migrated to AWS
Save time & cost—migrate to fully managed databases

Explore Our Products



Berkeley Architecture Research

Opening an AWS Account



Sign up details

aws

English ▾

Create an AWS account

AWS Accounts Include
12 Months of Free Tier Access

Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB
Visit aws.amazon.com/free for full offer terms

Email address
* Email is a required field

Password

Confirm password

AWS account name

Continue

[Sign in to an existing AWS account](#)

© 2020 Amazon Web Services, Inc. or its affiliates.
All rights reserved.
[Privacy Policy](#) | [Terms of Use](#)



Opening an AWS Account



Additional sign up details

Make sure to pick a **Personal** account

Contact Information

All fields are required.

Please select the account type and complete the fields below with your contact details.

Account type

Professional Personal

Full name

* Full Name is a required field.

Phone number

Country/Region

United States

Address

Street, P.O. Box, Company Name, c/o
 Apartment, suite, unit, building, floor, etc.

City

State / Province or region

Postal code

Check here to indicate that you have read and agree to the terms of the [AWS Customer Agreement](#)

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy | Terms of Use | Sign Out





Opening an AWS Account

Payment details:

If you plan on using AWS in the future for other projects (personal or otherwise), you should probably sign up with your credit card. Otherwise....

The screenshot shows the 'Credit/Debit card number' field, an 'Expiration date' dropdown set to '02 2020', and a 'Cardholder's name' field. Below these, there are two radio button options: 'Use my contact address' (selected) and 'Use a new address'. The 'Use a new address' section includes fields for 'Full name' (with a required field error message), 'Phone number', 'Country/Region' (set to 'United States'), 'Address' (with two sub-fields for street and suite/building), 'City', 'State / Province or region', and 'Postal code'. A 'Verify and Add' button is at the bottom.

Credit/Debit card number

Expiration date
02 2020

Cardholder's name

Billing address

Use my contact address

Use a new address

Full name

* Full Name is a required field.

Phone number

Country/Region
United States

Address

Street, P.O. Box, Company Name, c/o

Apartment, suite, unit, building, floor, etc.

City

State / Province or region

Postal code

Verify and Add





Opening an AWS Account

Standard confirmation stuff...

And that should be it for signing up!

aws English ▾

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

Text message (SMS) Voice call

Country or region code

Cell Phone Number

Security check

Type the characters as shown above

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy | Terms of Use | Sign Out



Opening an AWS Account



Log in to your AWS console

The screenshot shows the AWS homepage with a red box highlighting the "Sign In to the Console" button in the top right corner. The page includes the AWS logo, navigation links like Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search bar. The main content area displays a welcome message: "Welcome to Amazon Web Services" and "Thank you for creating an Amazon Web Services Account. We are activating your account, which should only take a few minutes. You will receive an email when this is complete." Below this, there's a "Personalize Your Experience" section with dropdown menus for "My role is:" and "I am interested in:", both currently set to "select role" and "select area". There are also "Submit" and "Contact Sales" buttons.

Try AWS with a 10-Minute Tutorial



Launch a Linux Virtual Machine



Store Your Files in the Cloud



Launch a WordPress Website



Launch a Web Application



Opening an AWS Account



Make sure your region
is set to US East (North
Virginia), a.k.a us-east-1

The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a search icon. To the right of the search icon, the region 'N. Virginia' is selected, indicated by a red box. Below the navigation bar, the title 'AWS Management Console' is displayed. On the left, there's a sidebar titled 'AWS services' with a 'Find Services' search bar and a 'All services' link. The main content area is titled 'Build a solution' and features eight quick-launch cards: 'Launch a virtual machine' (With EC2, 2-3 minutes, icon of a CPU), 'Build a web app' (With Elastic Beanstalk, 6 minutes, icon of a cloud with a key), 'Build using virtual servers' (With Lightsail, 1-2 minutes, icon of a server with a speaker), 'Register a domain' (With Route 53, 3 minutes, icon of a shield with the number 53), 'Connect an IoT device' (With AWS IoT, 5 minutes, icon of a circular device), 'Start migrating to AWS' (With CloudEndure Migration, 1-2 minutes, icon of a cloud with an arrow), 'Start a development project' (With CodeStar, 5 minutes, icon of a person working on a laptop), and 'Deploy a serverless microservice' (With Lambda, API Gateway, 2 minutes, icon of two interlocking boxes). At the bottom of this section is a 'See more' link. To the right of the main content area is a sidebar titled 'Access resources on' with a mobile phone icon, 'Explore AWS' with a globe icon, 'Amazon GuardDuty' with a shield icon, 'Amazon SageMaker Studio' with a monitor icon, 'AWS IQ' with a brain icon, 'Free Digital Training' with a graduation cap icon, and 'Have feedback?' with an envelope icon. The sidebar also lists various AWS regions: US East (N. Virginia) us-east-1, US East (Ohio) us-east-2, US West (N. California) us-west-1, US West (Oregon) us-west-2, Asia Pacific (Hong Kong) ap-east-1, Asia Pacific (Mumbai) ap-south-1, Asia Pacific (Seoul) ap-northeast-2, Asia Pacific (Singapore) ap-southeast-1, Asia Pacific (Sydney) ap-southeast-2, Asia Pacific (Tokyo) ap-northeast-1, Canada (Central) ca-central-1, Europe (Frankfurt) eu-central-1, Europe (Ireland) eu-west-1, Europe (London) eu-west-2, Europe (Paris) eu-west-3, Europe (Stockholm) eu-north-1, Middle East (Bahrain) me-south-1, and South America (São Paulo) sa-east-1. There's also a link to 'Submit feedback'.



FireSim Docs – AWS User Setup



If you are affiliated with a lab that has access to centralized AWS credits, we recommend checking with your lab about associating your account with the lab's central AWS payment pool

Docs » 2. Initial Setup/Installation » 2.1. First-time AWS User Setup [Edit on GitHub](#)

2.1. First-time AWS User Setup

If you've never used AWS before and don't have an account, follow the instructions below to get started.

2.1.1. Creating an AWS Account

First, you'll need an AWS account. Create one by going to aws.amazon.com and clicking "Sign Up." You'll want to create a personal account. You will have to give it a credit card number.

2.1.2. AWS Credit at Berkeley

If you're an internal user at Berkeley and affiliated with UCB-BAR or the RISE Lab, see the [RISE Lab Wiki](#) for instructions on getting access to the AWS credit pool. Otherwise, continue with the following section.

2.1.3. Requesting Limit Increases

In our experience, new AWS accounts do not have access to EC2 F1 instances by default. In order to get access, you should file a limit increase request. You can learn more about EC2 instance limits here: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html#ec2-on-demand-instances-limits>

To request a limit increase, follow these steps:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html>

You'll probably want to start out with the following request, depending on your existing limits:

Limit Type:	EC2 Instances
-------------	---------------



FireSim Docs – AWS User Setup



We will be using AWS F1 FPGA instances, which are not a “common” type of instance. Therefore, we need to request a “limit increase” for this type of instance. This process has a human-in-the-loop, so it might take a few hours up to a couple of days

FireSim
latest

Search docs

GETTING STARTED:

- 1. FireSim Basics
- 2. Initial Setup/Installation
 - 2.1. First-time AWS User Setup
 - 2.1.1. Creating an AWS Account
 - 2.1.2. AWS Credit at Berkeley
 - 2.1.3. Requesting Limit Increases
 - 2.2. Configuring Required Infrastructure in Your AWS Account
 - 2.3. Setting up your Manager Instance
- 3. Running FireSim Simulations
- 4. Building Your Own Hardware Designs (FireSim FPGA Images)

ADVANCED DOCS:

- Manager Usage (the `firesim` command)
- Workloads
- Targets
- Debugging
- Supernode - Multiple Simulated SoCs Per FPGA
- Miscellaneous Tips
- FireSim Asked Questions

GOLDEN GATE (MIDAS II) DOCS:

Read the Docs v: latest ▾

Docs » 2. Initial Setup/Installation » 2.1. First-time AWS User Setup [Edit on GitHub](#)

2.1. First-time AWS User Setup

If you've never used AWS before and don't have an account, follow the instructions below to get started.

2.1.1. Creating an AWS Account

First, you'll need an AWS account. Create one by going to aws.amazon.com and clicking "Sign Up." You'll want to create a personal account. You will have to give it a credit card number.

2.1.2. AWS Credit at Berkeley

If you're an internal user at Berkeley and affiliated with UCB-BAR or the RISE Lab, see the [RISE Lab Wiki](#) for instructions on getting access to the AWS credit pool. Otherwise, continue with the following section.

2.1.3. Requesting Limit Increases

In our experience, new AWS accounts do not have access to EC2 F1 instances by default. In order to get access, you should file a limit increase request. You can learn more about EC2 instance limits here: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html#ec2-on-demand-instances-limits>

To request a limit increase, follow these steps:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html>

You'll probably want to start out with the following request, depending on your existing limits:

Limit Type:	EC2 Instances
Current Value:	1
New Value:	10

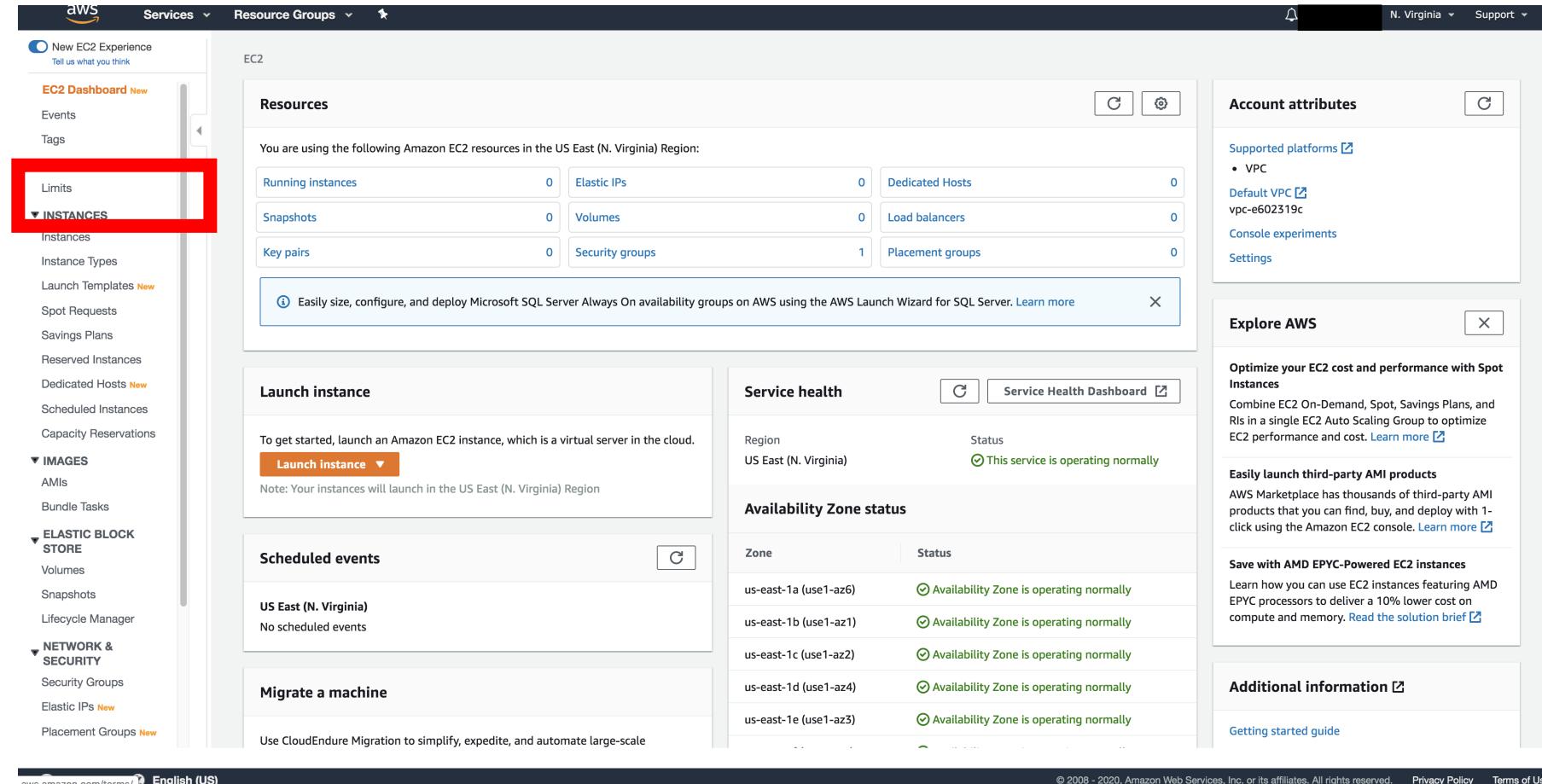




AWS F1 Instance Limit Increase

Access your EC2 dashboard
at
<https://console.aws.amazon.com/ec2>

Select “Limits” from the
menu on the left



The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'INSTANCES' section, the 'Limits' option is highlighted with a red box. The main content area displays various EC2 resources and their counts: Running instances (0), Elastic IPs (0), Dedicated Hosts (0), Snapshots (0), Volumes (0), Load balancers (0), Key pairs (0), Security groups (1), and Placement groups (0). Below this, there are sections for 'Launch instance', 'Scheduled events' (showing 'US East (N. Virginia)' with 'No scheduled events'), and 'Migrate a machine'. To the right, there are sections for 'Service health' (status: 'This service is operating normally') and 'Availability Zone status' (listing zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, all marked as 'Availability Zone is operating normally'). The top right corner shows 'N. Virginia' and 'Support'.





AWS F1 Instance Limit Increase

Select “Request limit increase” on the right side of the window

This will open a new support case with AWS

The screenshot shows the AWS EC2 Limits page. On the left is a navigation sidebar with links like New EC2 Experience, EC2 Dashboard, Events, Tags, Reports, and Limits. The main content area is titled "Limits" and contains a table with columns: Name, Limit type, Current limit, and Description. The first item in the table is "Launch configurations" with an "Auto Scaling" limit type and a current limit of 200. A red box highlights the "Request limit increase" button located at the top right of the table header.

Name	Limit type	Current limit	Description
Launch configurations	Auto Scaling	200	The maximum number of launch configurations for your account.
Auto Scaling groups	Auto Scaling	200	The maximum number of Auto Scaling groups for your account.
Running M5N Dedicated Hosts	Dedicated Hosts	0	-
Running C3 Dedicated Hosts	Dedicated Hosts	0	-
Running C4 Dedicated Hosts	Dedicated Hosts	0	-
Running C5 Dedicated Hosts	Dedicated Hosts	0	-
Running INF1 Dedicated Hosts	Dedicated Hosts	0	-
Running Z1D Dedicated Hosts	Dedicated Hosts	0	-
Running D2 Dedicated Hosts	Dedicated Hosts	0	-
Running F1 Dedicated Hosts	Dedicated Hosts	0	-
Running G2 Dedicated Hosts	Dedicated Hosts	0	-
Running G3 Dedicated Hosts	Dedicated Hosts	0	-
Running H1 Dedicated Hosts	Dedicated Hosts	0	-
Running X1E Dedicated Hosts	Dedicated Hosts	0	-
Running I2 Dedicated Hosts	Dedicated Hosts	0	-
Running I3 Dedicated Hosts	Dedicated Hosts	0	-
Running G3S Dedicated Hosts	Dedicated Hosts	0	-





AWS F1 Instance Limit Increase

Select the Service limit increase case type.

For Case classification, select “EC2 Instances”

The screenshot shows the AWS Support Center interface for creating a new support case. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a bell icon, 'Global', and 'Support'. The main content area is titled 'Create case' and shows three options:

- Account and billing support**: Assistance with account and billing-related enquiries.
- Service limit increase**: Requests to increase the service limit of your AWS resources. This option is highlighted with a red box.
- Technical support**: Service-related technical issues and third-party applications. This option is also highlighted with a red box.

Below these options is a 'Case classification' section, also highlighted with a red box. It contains:

- Limit type**: A dropdown menu showing 'EC2 Instances'.
- Severity**: A dropdown menu showing 'Info'.

On the right side of the 'Case classification' section is a 'Useful links' box containing links to 'Amazon EC2 Service Limits', 'Amazon EC2 On-Demand Instance limits', and 'vCPU-based On-Demand Instance Limits FAQ'.

Further down the page is a 'Requests' section with a note: "To request additional limit increases for the same limit type, choose Add another request. To request an increase for a different limit type, create a separate limit increase request." Below this is a 'Request 1' section with a 'Region' dropdown set to 'US East (Northern Virginia)' and a 'Remove' button.

The bottom of the page features a footer with links for 'Feedback', 'English (US)', and 'Privacy Policy'.





AWS F1 Instance Limit Increase

Select the following options:

Region: **US East (Northern Virginia)**

Primary Instance Type: **All F Instances**

Limit: **Instance Limit**

New limit value: **64**

Screenshot of the AWS Support Case Classification interface for requesting an instance limit increase.

The "Case classification" section shows:

- Limit type: EC2 Instances
- Severity: General question
- Useful links:
 - Amazon EC2 Service Limits
 - Amazon EC2 On-Demand Instance limits
 - vCPU-based On-Demand Instance Limits FAQ

The "Requests" section shows:

- A note: "To request additional limit increases for the same limit type, choose Add another request. To request an increase for a different limit type, create a separate limit increase request."
- A "Request 1" card with the following fields:
 - Region: US East (Northern Virginia)
 - Primary Instance Type: All F instances
 - Limit: Instance Limit
 - New limit value: 64

At the bottom of the page, there are footer links: Feedback, English (US), Copyright notice (2008-2020), Privacy Policy, and Terms of Use.





AWS F1 Instance Limit Increase

Use case description:

Class project for UC Berkeley EE 290-2 (Hardware for Machine Learning). The class project uses FireSim (<https://fires.im>) on AWS F1 instances.

The screenshot shows the AWS Support interface for creating a new case. The 'Case description' section contains an 'Assistance' box with information about EC2 On-Demand Instance limits. Below it is a text input field for 'Use case description' containing the text: "Class project for UC Berkeley EE 290-2 (Hardware for Machine Learning). The class project uses FireSim (<https://fires.im>) on AWS F1 instances." This input field is highlighted with a red border. The 'Contact options' section below includes fields for preferred contact language (set to English) and contact methods (Web selected). At the bottom are 'Cancel' and 'Submit' buttons, along with links for Feedback, Language (English (US)), and legal notices (Privacy Policy, Terms of Use).





AWS F1 Instance Limit Increase

This will open an AWS support case. You should receive a few confirmation emails.

Screenshot of the AWS Support Center showing a new support case for an EC2 instance limit increase.

Case details:

Subject	Status
Limit Increase: EC2 Instances	Unassigned
Case ID	Severity
6827780361	General question
Created	Category
2020-02-21T23:17:12.599Z	Service Limit Increase, EC2 Instances
Case type	Additional contacts
Service limits	-
Opened by	
hngenc@berkeley.edu	

Correspondence:

hngenc	Limit increase request 1 Service: EC2 Instances Region: US East (Northern Virginia) Primary Instance Type: All F instances Limit name: Instance Limit New limit value: 64 ----- Use case description: Class project for UC Berkeley EE 290-2 (Hardware for Machine Learning). The class project uses FireSim (https://firesim.org) on AWS F1 instances.
--------	--





AWS F1 Instance Limit Increase

One of those emails will tell you the request requires further internal review. This should hopefully not take more than a day.

We are asking up-to 64 F1 instances but if you get less (~16) that should be fine

RE: [CASE 6827780361] Limit Increase: EC2 Instances Inbox x

 no-reply-aws@amazon.com <no-reply-aws@amazon.com>

to me ▾

3:21 PM (4 minutes ago)



Hello!

I am following up to notify you that we've received your limit increase request.

I see that you have requested the following:

[US_EAST_1]: EC2 Instances / Instance Limit (All F instances), New Limit = 64

This specific limit increase request requires further internal review before approval and I have initiated that review.

I understand this increase is important to you and I will do my best to get you a prompt answer.

I will notify you as soon as I have an update. Thank you for your patience while I work with our Service Team.

To contact us again about this case, please return to the AWS Support Center using the following URL:

<https://console.aws.amazon.com/support/home#/case/?displayId=6827780361&language=en>

(If you are connecting by federation, log in before following the link.)

*Please note: this e-mail was sent from an address that cannot accept incoming e-mail. Please use the link above if you need to contact us again about this same issue.

=====
Learn to work with the AWS Cloud. Get started with free online videos and self-paced labs at
<http://aws.amazon.com/training/>
=====

Amazon Web Services, Inc. is an affiliate of Amazon.com, Inc. Amazon.com is a registered trademark of Amazon.com, Inc. or its affiliates.



Lab 3 Instructions – Class AWS Form



Tell us about your AWS account through a Google Form

EE 290-2 Spring 2020

Lab 3: Tiling and Optimization for Accelerators

1 Introduction

This lab will provide you with hands-on experience on the implications of mapping large matrix multiplication operations onto 2D systolic array accelerators, and give you experience using the Amazon Web Services (AWS) EC2 public cloud for FPGA-accelerated simulation.

As most neural network models do not fit in on-chip memory, loop blocking/tiling is an important tool in writing a performant neural network implementation. The additional degree of freedom afforded by the scratchpad in ML accelerators requires further planning of data re-use within the scratchpad. Furthermore, the size of the compute array adds an additional constraint on the tiling hierarchy.

There is a wide body of literature on loop blocking and scheduling for standard scalar processors. However, this body of literature is less extensive with regards to on-chip accelerators with dedicated memories that may be connected to the memory hierarchy in various forms.

Specifically, the DMA of the Gemmini accelerator is currently connected to the shared L2 cache of the scalar host processor. As such, it may see caching affects from the tiling scheme, as well as other parts of the program.

In this lab you will continue using the Chipyard and Gemmini platforms from the previous lab to further improve the performance of DNN execution using ML accelerators through software optimization. You will also be using the FireSim platform (within Chipyard) on the AWS EC2 public cloud.

1.1 FireSim and Amazon Web Services (AWS)

In order to use FireSim on AWS (you will find additional information about FireSim in the next section), you will need to open an AWS account. AWS is a paid service, so be careful about your usage (more details to follow). Do not wait with this process until the last minute, because it might have around a 1-day latency due to humans-in-the-loop. The lecture on Wednesday, February 26th, will provide a step-by-step tutorial describing the procedures in this section. We recommend following the

When you are done opening your AWS account, please fill out [this form](#)². After you fill out the form, you will receive an email from us with a \$200 AWS credit promo code. This amount should be more than sufficient for the tasks in this assignment, assuming you manage your resources and track your AWS expenses. In addition, you can receive another promo code of \$100 by signing up as a student in [AWS Educate](#)³ with your [berkeley.edu](#) email address. Do not sign up for the AWS Educate Starter Kit, because you will not have access to the FPGA-based F1 instances which are required for this lab. You will need to redeem your AWS promo code credits by following the [instructions](#)⁴.

FireSim AWS infrastructure setup instructions in [this page](#)⁵ of the FireSim documentation. Do not continue to setting up your manager instance (we will provide you with a prepared manager AMI to shorten some of the build preparation process).

2 Background

2.1 Loop Tiling and Scheduling

We will be working on tiling matrix multiplication, since Gemmini accelerates matrix multiplication operations, and we saw in the previous lab that convolutions can be lowered to matrix multiplication operations. As a reminder, a standard nested-loop matrix multiplication operation looks as follows:

¹<https://docs.fires.im/en/latest/Initial-Setup/First-time-AWS-User-Setup.html>

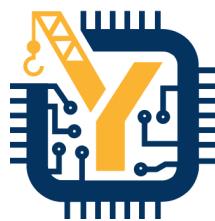
²<https://forms.gle/YLJKrgLGpdzpuqk59>

³<https://aws.amazon.com/education/awseducate/>

⁴<https://aws.amazon.com/awscredits/>

⁵<https://docs.fires.im/en/latest/Initial-Setup/Configuring-Required-Infrastructure-in-Your-AWS-Account.html>





Tell Us About Your AWS Account

Fill the form at

<https://forms.gle/ALvSrq9NK32xmWXy9>

We need you to tell us about your AWS account for two reasons:

1. So we can send you a promo code for \$200 credits
2. So we can share with you the prebuild AMI (image)

EE290-2 Hardware for Machine Learning AWS Accounts

Please fill out the details of your AWS account, so we can send you a promo code with \$200 credits to use in the class, as well as share the class AMI with you.

Name
Your answer

Student ID
Your answer

Email
Your answer

AWS Account Number. Find this 12-digit number by going to your account settings page in your AWS console, or through the support center (https://docs.aws.amazon.com/AM/latest/UserGuide/console_account-alias.html#FindingYourAWSId)
Your answer

Submit

Never submit passwords through Google Forms.
This form was created inside of UC Berkeley. Report Abuse





Tell Us About Your AWS Account

Within a day (hopefully less), you will get an email from us with an AWS promo code

EE290 AWS Promo Code Inbox ×

 **Alon Amid**
to me ▾

Hello Hasan

This is your AWS Promo code:

PC10UOS07VQN4O2

Best,
EE290 Staff

Reply Forward



Redeem AWS Credits



In your AWS billing console/dashboard, select Credits, fill the promo code, and Redeem

The screenshot shows the AWS Billing Credits page. On the left, a sidebar lists various billing-related options, with 'Credits' highlighted by a red box. The main content area has a heading 'Credits' and a sub-instruction 'Please enter your code below to redeem your credits.' It includes a 'Promo Code' input field, a CAPTCHA section with a challenge image ('4afgnp') and a 'Refresh Image' link, and a text input for entering the CAPTCHA characters. Below these is a checkbox for accepting terms and conditions, followed by a large blue 'Redeem' button. At the bottom, there's a note about the table displaying redeemed credits and a message indicating no credits are currently available.

Please enter your code below to redeem your credits.

Promo Code

Security Check  Refresh Image

Please type the characters as shown above

By clicking "Redeem" you indicate that you have read and agree to the terms of the AWS Promotional Credit Terms & Conditions located [here](#).

Redeem

The table below displays all AWS credits redeemed by your account. Credits are automatically applied to charges associated with qualifying AWS service usage. Please note that the values for used and remaining credit amounts are updated each month when your invoice is finalized.

You currently have no redeemable credits.

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Lab 3 Instructions –FireSim Infra. Setup



Once you have
redeemed your credits,
continue setting up the
FireSim infrastructure

EE 290-2 Spring 2020

Lab 3: Tiling and Optimization for Accelerators

1 Introduction

This lab will provide you with hands-on experience on the implications of mapping large matrix multiplication operations onto 2D systolic array accelerators, and give you experience using the Amazon Web Services (AWS) EC2 public cloud for FPGA-accelerated simulation.

As most neural network models do not fit in on-chip memory, loop blocking/tiling is an important tool in writing a performant neural network implementation. The additional degree of freedom afforded by the scratchpad in ML accelerators requires further planning of data re-use within the scratchpad. Furthermore, the size of the compute array adds an additional constraint on the tiling hierarchy.

There is a wide body of literature on loop blocking and scheduling for standard scalar processors. However, this body of literature is less extensive with regards to on-chip accelerators with dedicated memories that may be connected to the memory hierarchy in various forms.

Specifically, the DMA of the Gemmini accelerator is currently connected to the shared L2 cache of the scalar host processor. As such, it may see caching affects from the tiling scheme, as well as other parts of the program.

In this lab you will continue using the Chipyard and Gemmini platforms from the previous lab to further improve the performance of DNN execution using ML accelerators through software optimization. You will also be using the FireSim platform (within Chipyard) on the AWS EC2 public cloud.

1.1 FireSim and Amazon Web Services (AWS)

In order to use FireSim on AWS (you will find additional information about FireSim in the next section), you will need to open an AWS account. AWS is a paid service, so be careful about your usage (more details to follow). Do not wait with this process until the last minute, because it might have around a 1-day latency due to humans-in-the-loop. The lecture on Wednesday, February 26th, will provide a step-by-step tutorial describing the procedures in this section. We recommend following the instructions in [this page](#)¹ of the FireSim documentation.

When you are done opening your AWS account, please fill out [this form](#)². After you fill out the form, you will receive an email from us with a \$200 AWS credit promo code. This amount should be more than sufficient for the tasks in this assignment, assuming you manage your resources and track your AWS expenses. In addition, you can receive another promo code of \$100 by signing up as a student in [AWS Educate](#)³ with your [berkeley.edu](#) email address. Do not sign up for the AWS Educate Starter Kit, because you will not have access to the FPGA-based F1 instances which are required for this lab. You will need to redeem your AWS promo code credits by following the [instructions](#)⁴.

Once you have opened an AWS account and applied your promo code for credits, follow the initial FireSim AWS infrastructure setup instructions in [this page](#)⁵ of the FireSim documentation. Do not continue to setting up your manager instance (we will provide you with a prepared manager AMI to shorten some of the build preparation process).

2 Background

2.1 Loop Tiling and Scheduling

We will be working on tiling matrix multiplication, since Gemmini accelerates matrix multiplication operations, and we saw in the previous lab that convolutions can be lowered to matrix multiplication operations. As a reminder, a standard nested-loop matrix multiplication operation looks as follows:

¹<https://docs.firesim.im/en/latest/Initial-Setup/First-time-AWS-User-Setup.html>

²<https://forms.gle/YLJKrgLGpdzpupk59>

³<https://aws.amazon.com/education/awseducate/>

⁴<https://aws.amazon.com/awscredits/>

⁵<https://docs.firesim.im/en/latest/Initial-Setup/Configuring-Required-Infrastructure-in-Your-AWS-Account.html>



FireSim Docs – Configuring Infrastructure



“Select a region”: we already did this (us-east-1, Northern Virginia), remember?

“Key setup”: go back to your EC2 console

The screenshot shows the left sidebar of the FireSim documentation. The sidebar has a blue header with the title 'FireSim' and 'latest'. Below it is a search bar labeled 'Search docs'. The main navigation menu includes sections like 'GETTING STARTED:', '1. FireSim Basics', '2. Initial Setup/Installation' (which is expanded), '2.1. First-time AWS User Setup', '2.2. Configuring Required Infrastructure in Your AWS Account' (which is also expanded), '2.2.1. Select a region', '2.2.2. Key Setup', '2.2.3. Check your EC2 Instance Limits', '2.2.4. Start a t2.nano instance to run the remaining configuration commands', '2.2.5. Run scripts from the t2.nano', '2.2.6. Terminate the t2.nano', '2.2.7. Subscribe to the AWS FPGA Developer AMI', '2.3. Setting up your Manager Instance', '3. Running FireSim Simulations', '4. Building Your Own Hardware Designs (FireSim FPGA Images)', 'ADVANCED DOCS:', 'Manager Usage (the `firesim` command)', and 'Workloads'. At the bottom of the sidebar are links for 'Read the Docs' and 'v: latest'.

Docs » 2. Initial Setup/Installation » 2.2. Configuring Required Infrastructure in Your AWS Account

[Edit on GitHub](#)

2.2. Configuring Required Infrastructure in Your AWS Account

Once we have an AWS Account setup, we need to perform some advance setup of resources on AWS. You will need to follow these steps even if you already had an AWS account as these are FireSim-specific.

2.2.1. Select a region

Head to the [EC2 Management Console](#). In the top right corner, ensure that the correct region is selected. You should select one of: `us-east-1` (N. Virginia), `us-west-2` (Oregon), or `eu-west-1` (Ireland), since F1 instances are only available in those regions.

Once you select a region, it's useful to bookmark the link to the EC2 console, so that you're always sent to the console for the correct region.

2.2.2. Key Setup

In order to enable automation, you will need to create a key named `firesim`, which we will use to launch all instances (Manager Instance, Build Farm, Run Farm).

To do so, click “Key Pairs” under “Network & Security” in the left-sidebar. Follow the prompts, name the key `firesim`, and save the private key locally as `firesim.pem`. You can use this key to access all instances from your local machine. We will copy this file to our manager instance later, so that the manager can also use it.

2.2.3. Check your EC2 Instance Limits

AWS limits access to particular instance types for new/infrequently used accounts to protect their



Key Setup



Select “Key Pairs” on the left menu.

Then select “Create key pair” on the top right

The screenshot shows the AWS EC2 console. The left sidebar has a tree view of services: Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, IMAGES (AMIs, Bundle Tasks), ELASTIC BLOCK STORE (Volumes, Snapshots, Lifecycle Manager), NETWORK & SECURITY (Security Groups, Elastic IP), Placement Groups, Key Pairs (highlighted with a red box), and Network Interfaces. The main content area is titled "Key pairs" and shows a table with columns for Name and Fingerprint. A message at the bottom says "No key pairs to display". In the top right corner of the main area, there is a "Create key pair" button, which is also highlighted with a red box. The top navigation bar includes links for "Services", "Resource Groups", "Actions", and "Create key pair". The status bar at the bottom shows "Feedback", "English (US)", and copyright information: "© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use".



Key Setup



Name the key “firesim”,
and make sure it’s in
“pem” format.

Download the key, and
**make sure you keep it in
a safe place, and back it
up.**

Screenshot of the AWS EC2 "Create key pair" wizard:

The "Name" field contains "firesim". The "File format" section has "pem" selected (radio button is checked). The "Create key pair" button is visible at the bottom right.

Key pair details:
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name: firesim

File format:

pem
For use with OpenSSH

ppk
For use with PuTTY

Cancel Create key pair

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



FireSim Docs – Configuring Infrastructure



“Check your EC2 Instance Limits”: We’ve already requested a limit increase. Hopefully, it was approved by now.

Start a t2.nano instance

- 2.1. First-time AWS User Setup
- 2.2. Configuring Required Infrastructure in Your AWS Account
 - 2.2.1. Select a region
 - 2.2.2. Key Setup
 - 2.2.3. Check your EC2 Instance Limits
 - 2.2.4. Start a t2.nano instance to run the remaining configuration commands
 - 2.2.5. Run scripts from the t2.nano
 - 2.2.6. Terminate the t2.nano
 - 2.2.7. Subscribe to the AWS FPGA Developer AMI
- 2.3. Setting up your Manager Instance
- 3. Running FireSim Simulations
- 4. Building Your Own Hardware Designs (FireSim FPGA Images)

ADVANCED DOCS:

- Manager Usage (the `firesim` command)
- Workloads
- Targets
- Debugging
- Supernode - Multiple Simulated SoCs Per FPGA
- Miscellaneous Tips
- FireSim Asked Questions

GOLDEN GATE (MIDAS II) DOCS:

- Overview & Philosophy

[Read the Docs](#) v: latest ▾

2.2.3. Check your EC2 Instance Limits

AWS limits access to particular instance types for new/infrequently used accounts to protect their infrastructure. You should make sure that your account has access to `f1.2xlarge`, `f1.4xlarge`, `f1.16xlarge`, `m4.16xlarge`, and `c5.4xlarge` instances by looking at the “Limits” page in the EC2 panel, which you can access [here](#). The values listed on this page represent the maximum number of any of these instances that you can run at once, which will limit the size of simulations (# of nodes) that you can run. If you need to increase your limits, follow the instructions on the [Requesting Limit Increases](#) page. To follow this guide, you need to be able to run one `f1.2xlarge` instance and two `c5.4xlarge` instances.

2.2.4. Start a t2.nano instance to run the remaining configuration commands

To avoid having to deal with the messy process of installing packages on your local machine, we will spin up a very cheap `t2.nano` instance to run a series of one-time aws configuration commands to setup our AWS account for FireSim. At the end of these instructions, we'll terminate the `t2.nano` instance. If you happen to already have `boto3` and the AWS CLI installed on your local machine, you can do this locally.

Launch a `t2.nano` by following these instructions:

1. Go to the [EC2 Management Console](#) and click “Launch Instance”
2. On the AMI selection page, select “Amazon Linux AMI...”, which should be the top option.
3. On the Choose an Instance Type page, select `t2.nano`.
4. Click “Review and Launch” (we don’t need to change any other settings)
5. On the review page, click “Launch”
6. Select the `firesim` key pair we created previously, then click Launch Instances.
7. Click on the instance name and note its public IP address.

2.2.5. Run scripts from the t2.nano

SSH into the `t2.nano` like so:



Configuring FireSim AWS Infrastructure



Select “Instances” on the left menu.

Then select “Launch Instance”

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with several sections: Instances (with 'Instances' highlighted), Images, Elastic Block Store, and Network & Security. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and a 'Launch Instance' button. The main content area displays a message: 'You do not have any running instances in this region.' It also says 'First time using EC2? Check out the [Getting Started Guide](#). Click the Launch Instance button to start your own server.' A large blue 'Launch Instance' button is located in the bottom right of the main content area. The top right of the slide shows the AWS navigation bar with 'N. Virginia' and 'Support' dropdowns, along with some icons.



Configuring FireSim AWS Infrastructure



Select the “Amazon Linux 2 AMI”

The screenshot shows the AWS CloudFormation 'Create New Stack' wizard, Step 1: Choose an Amazon Machine Image (AMI). The 'Amazon Linux 2 AMI (HVM), SSD Volume Type' is selected and highlighted with a red box. The 'Select' button next to it is also highlighted.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start	AMI Name	Description	Root device type	Virtualization type	ENAs Enabled	Select
My AMIs	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0a887e401f7654935 (64-bit x86) / ami-002cc39e7bf021a77 (64-bit Arm)	Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.	ebs	hvm	Yes	<input checked="" type="button"/> Select 64-bit (x86) 64-bit (Arm)
AWS Marketplace	Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0e2ff28bb72a4e45	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	ebs	hvm	Yes	<input type="button"/> Select 64-bit (x86) 64-bit (Arm)
Community AMIs	Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0c322300a1dd5dc79 (64-bit x86) / ami-03587fa4048e9eb92 (64-bit Arm)	Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	ebs	hvm	Yes	<input type="button"/> Select 64-bit (x86) 64-bit (Arm)
Free tier only	SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-0df6cfabfbe4385b7 (64-bit x86) / ami-0e83525f58b2878f0 (64-bit Arm)	SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.	ebs	hvm	Yes	<input type="button"/> Select 64-bit (x86) 64-bit (Arm)
	Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-07ebfd5b3428b6f4d (64-bit x86) / ami-0400a1104d5b9caa1 (64-bit Arm)	Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).	ebs	hvm	Yes	<input type="button"/> Select 64-bit (x86) 64-bit (Arm)
	Amazon RDS	Are you launching a database instance? Try Amazon RDS.				<input type="button"/> Hide

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use





Configuring FireSim AWS Infrastructure

Select a t2.nano instance, and then click “Review and Launch”

Screenshot of the AWS EC2 Instance Creation Wizard - Step 2: Choose an Instance Type.

The screenshot shows a table of instance types. A red box highlights the checkbox for the first row (t2.nano). Another red box highlights the "Review and Launch" button at the bottom right.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.nano (Variable ECUs, 1 vCPUs, 2.4 GHz, Intel Xeon Family, 0.5 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input checked="" type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Next: Configure Instance Details **Review and Launch**



Configuring FireSim AWS Infrastructure



Review the details, and Launch.

It should then prompt you to select a key pair

Screenshot of the AWS Step 7: Review Instance Launch page. The page shows configuration details for launching an instance:

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0e2ff28bfb72a4e45
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.nano	Variable	1	0.5	EBS only	-	Low to Moderate

Security Groups
Security group name: launch-wizard-1
Description: launch-wizard-1 created 2020-02-21T15:33:30.149-08:00
This security group has no rules

Launch Button (highlighted with a red box)

Navigation links: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review.



Configuring FireSim AWS Infrastructure



Choose “Choose and existing key pair”, and then select the “firesim” key pair that you created previously

Select an existing key pair or create a new key pair X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

firesim

I acknowledge that I have access to the selected private key file (firesim.pem), and that without this file, I won't be able to log into my instance.

Cancel **Launch Instances**



Configuring FireSim AWS Infrastructure



You will see a confirmation screen that your instance has launched

AWS Services Resource Groups ⚙️ 🔔 N. Virginia Support

Launch Status

Your instances are now launching
The following instance launches have been initiated: i-0f838d80012f12b74 [View launch log](#)

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

[View Instances](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Configuring FireSim AWS Infrastructure



You will also now see the detail of your instance in the instances list, including the public IP address of your instance

The screenshot shows the AWS EC2 Dashboard with the Instances list. The 'IPv4 Public IP' column is highlighted with a red box. The table row for the instance 'i-0f838d80012f12b74' shows the following details:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Last Launch
	i-0f838d80012f12b74	t2.nano	us-east-1d	running	Initializing	None	ec2-18-212-98-137.compute-1.amazonaws.com	18.212.98.137	-	firesim	disabled	February 1, 2020

The detailed view for the selected instance 'i-0f838d80012f12b74' shows the following configuration:

Description	Value
Instance ID	i-0f838d80012f12b74
Instance state	running
Instance type	t2.nano
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS	ip-172-31-16-164.ec2.internal
Private IPs	172.31.16.164
Secondary private IPs	vpc-e602319c
Public DNS (IPv4)	ec2-18-212-98-137.compute-1.amazonaws.com
IPv4 Public IP	18.212.98.137
IPv6 IPs	-
Elastic IPs	-
Availability zone	us-east-1d
Security groups	launch-wizard-1, view inbound rules, view outbound rules
Scheduled events	No scheduled events
AMI ID	amzn-ami-hvm-2018.03.0.20200206.0-x86_64-gp2 (ami-0e2ff28bf72a4e45)



FireSim Docs – Configuring Infrastructure



We now need to run the setup script on the t2.nano

- 1. FireSim Basics
- 2. Initial Setup/Installation
 - 2.1. First-time AWS User Setup
 - 2.2. Configuring Required Infrastructure in Your AWS Account
 - 2.2.1. Select a region
 - 2.2.2. Key Setup
 - 2.2.3. Check your EC2 Instance Limits
 - 2.2.4. Start a t2.nano instance to run the remaining configuration commands
 - 2.2.5. Run scripts from the t2.nano
 - 2.2.6. Terminate the t2.nano
 - 2.2.7. Subscribe to the AWS FPGA Developer AMI
 - 2.3. Setting up your Manager Instance
- 3. Running FireSim Simulations
- 4. Building Your Own Hardware Designs (FireSim FPGA Images)

ADVANCED DOCS:

- Manager Usage (the `firesim` command)
- Workloads
- Targets
- Debugging
- Supernode - Multiple Simulated SoCs Per FPGA
- Miscellaneous Tips
- FireSim Asked Questions

[Read the Docs](#) v: latest ▾

2.2.5. Run scripts from the t2.nano

SSH into the `t2.nano` like so:

```
ssh -i firesim.pem ec2-user@INSTANCE_PUBLIC_IP
```

Which should present you with something like:

```
Last login: Mon Feb 12 21:11:27 2018 from 136.152.143.34
[ec2-user@ip-172-30-2-66 ~]$
```

On this machine, run the following:

```
aws configure
[follow prompts]
```

See <https://docs.aws.amazon.com/cli/latest/userguide/tutorial-ec2-ubuntu.html#configure-cli-launch-ec2> for more about aws configure. Within the prompt, you should specify the same region that you chose above (one of `us-east-1`, `us-west-2`, `eu-west-1`) and set the default output format to `json`. You will need to generate an AWS access key in the “Security Credentials” menu of your AWS settings (as instructed in <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#access-keys-and-secret-access-keys>).

Again on the `t2.nano` instance, do the following:

```
sudo yum -y install python-pip
sudo pip install boto3
wget https://raw.githubusercontent.com/firesim/firesim/master/scripts/aws-setup.py
python aws-setup.py
```





Configuring FireSim AWS Infrastructure

```
$ chmod 400 firesim.pem  
$ ssh -i firesim.pem ec2-user@18.212.98.137
```

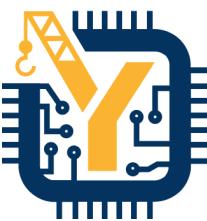
```
__|__|_)  
_||(_ /     Amazon Linux AMI  
____| \____|
```

```
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
4 package(s) needed for security, out of 10 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-16-164 ~]$ aws configure
```

Before running this command, we
need to setup AWS Access Keys



Configuring FireSim AWS Infrastructure



In your AWS EC2 console,
select “My Security
Credentials” from your
user menu on the top right
corner

The screenshot shows the AWS EC2 Dashboard. On the far right, there is a user menu with several options: 'My Account', 'My Organization', 'My Service Quotas', 'My Billing Dashboard', 'My Security Credentials' (which is highlighted with a red box), and 'Sign Out'. Below the menu, there's a section titled 'Explore AWS' with links to 'Easily launch third-party AMI products', 'Save with AMD EPYC-Powered EC2 instances', 'Optimize your EC2 cost and performance with Spot Instances', and 'Additional information'.

EC2 Dashboard

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Running instances	1	Elastic IPs	0	Dedicated Hosts	0
Snapshots	0	Volumes	1	Load balancers	0
Key pairs	1	Security groups	2	Placement groups	0

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#)

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#)

Note: Your instances will launch in the US East (N. Virginia) Region

Scheduled events

US East (N. Virginia)
No scheduled events

Migrate a machine

Use CloudEndure Migration to simplify, expedite, and automate large-scale migrations from physical, virtual, and cloud-based infrastructure to AWS.
[Get started with CloudEndure Migration](#)

Service health

Region: US East (N. Virginia) Status: This service is operating normally

Availability Zone status

Zone	Status
us-east-1a (use1-az6)	Availability Zone is operating normally
us-east-1b (use1-az1)	Availability Zone is operating normally
us-east-1c (use1-az2)	Availability Zone is operating normally
us-east-1d (use1-az4)	Availability Zone is operating normally
us-east-1e (use1-az3)	Availability Zone is operating normally
us-east-1f (use1-az5)	Availability Zone is operating normally

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



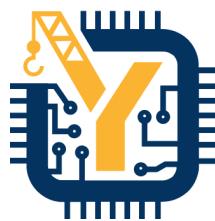
Configuring FireSim AWS Infrastructure



In the “Access keys” section, create a new access key

The screenshot shows the AWS Identity and Access Management (IAM) service in the AWS Management Console. The left sidebar lists various IAM management options. The main content area is titled "Your Security Credentials" and provides instructions for managing AWS credentials. It highlights the "Access keys (access key ID and secret access key)" section, which is where new access keys are created. A prominent blue button labeled "Create New Access Key" is highlighted with a red box. Below this button, a note states: "Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead." The bottom of the page includes standard AWS footer links for Feedback, English (US), and legal notices.





Configuring FireSim AWS Infrastructure

Download your root key file. This is a csv file that includes the Access Key ID, and the Secret Access Key (you will need both of them for the aws configure command)

SAVE THIS IN A SPECIAL LOCATION

Create Access Key ×

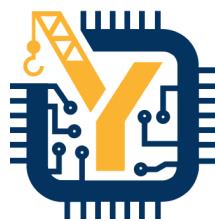
Your access key (access key ID and secret access key) has been created successfully.

Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

To help protect your security, store your secret access key securely and do not share it.

▶ Show Access Key Download Key File Close





Configuring FireSim AWS Infrastructure

Continue with the `aws configure` command, using your AWS Access Key.

For default region name, enter: **us-east-1**

For default output format, enter: **json**

```
[ec2-user@ip-172-31-16-164 ~]$ aws configure
AWS Access Key ID [None]: XXXXXXXX
AWS Secret Access Key [None]: XXXXXXXXXXXX
Default region name [None]: us-east-1
Default output format [None]: json
```





Configuring FireSim AWS Infrastructure

Run the setup scripts as instructed in the FireSim docs setup instructions:

```
sudo yum -y install python-pip  
sudo pip install boto3  
wget https://raw.githubusercontent.com/firesim/firesim/master/scripts/aws-setup.py  
python aws-setup.py  
...
```

```
Creating VPC for FireSim...  
Success!  
Creating a subnet in the VPC for each availability zone...  
Success!  
Creating a security group for FireSim...  
Success!
```



FireSim Docs – Configuring Infrastructure



To terminate the t2.nano instance, go back to your EC2 console, to the instances list

The sidebar includes the following sections:

- 2.2.2. Key Setup
- 2.2.3. Check your EC2 Instance Limits
- 2.2.4. Start a t2.nano instance to run the remaining configuration commands
- 2.2.5. Run scripts from the t2.nano
- 2.2.6. Terminate the t2.nano
- 2.2.7. Subscribe to the AWS FPGA Developer AMI

ADVANCED DOCS:

- 3. Running FireSim Simulations
- 4. Building Your Own Hardware Designs (FireSim FPGA Images)
- Manager Usage (the `firesim` command)
- Workloads
- Targets
- Debugging
- Supernode - Multiple Simulated SoCs Per FPGA
- Miscellaneous Tips
- FireSim Asked Questions

GOLDEN GATE (MIDAS II) DOCS:

- Overview & Philosophy
- Target Abstraction & Host Decoupling
- Target-to-Host Bridges
- Bridge Walkthrough

Read the Docs v: latest

that you chose above (one of `us-east-1`, `us-west-2`, `eu-west-1`) and set the default output format to `json`. You will need to generate an AWS access key in the “Security Credentials” menu of your AWS settings (as instructed in <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#access-keys-and-secret-access-keys>).

Again on the `t2.nano` instance, do the following:

```
sudo yum -y install python-pip
sudo pip install boto3
wget https://raw.githubusercontent.com/firesim/firesim/master/scripts/aws-setup.py
python aws-setup.py
```

This will create a VPC named `firesim` and a security group named `firesim` in your account.

2.2.6. Terminate the t2.nano

At this point, we are finished with the general account configuration. You should terminate the t2.nano instance you created, since we do not need it anymore (and it shouldn't contain any important data).

2.2.7. Subscribe to the AWS FPGA Developer AMI

Go to the [AWS Marketplace page for the FPGA Developer AMI](#). Click the button to subscribe to the FPGA Dev AMI (it should be free) and follow the prompts to accept the EULA (but do not launch any instances).

Now, hit next to continue on to setting up our Manager Instance.

Previous

Next

© Copyright 2018, Sagar Karandikar, Howard Mao, Donggyu Kim, David Biancolin, Alon Amid, and Berkeley Architecture Research. Revision 6a4b615d.

Built with [Sphinx](#) using a theme provided by [Read the Docs](#).



Configuring FireSim AWS Infrastructure



Right click on the instance entry in the table, and within the “Instance State” menu, select “Terminate”.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for Services, Resource Groups, and various EC2-related sections like Instances, Images, and Network & Security. The main area displays a table of EC2 instances. One instance, with Instance ID i-0f838d80012f12b74 and Public DNS ec2-18-212-98-137.compute-1.amazonaws.com, is selected. A context menu is open over this instance, with the "Terminate" option highlighted by a red box. Below the table, a detailed view of the selected instance is shown, including its description, status checks, monitoring, and tags. The instance is currently running (t2.nano). The bottom of the screen includes standard AWS footer links for Feedback, English (US), and various legal notices.



Configuring FireSim AWS Infrastructure



Confirm the instance termination

Terminate Instances

Warning
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

- i-0f838d80012f12b74 (ec2-18-212-98-137.compute-1.amazonaws.com)

Cancel **Yes, Terminate**



FireSim Docs – Configuring Infrastructure



Subscribe to the AWS
FPGA Developer AMI
(click on the link)

The sidebar includes the following links:
2.2.2. Key Setup
2.2.3. Check your EC2 Instance Limits
2.2.4. Start a t2.nano instance to run the remaining configuration commands
2.2.5. Run scripts from the t2.nano
2.2.6. Terminate the t2.nano
2.2.7. Subscribe to the AWS FPGA Developer AMI
3. Setting up your Manager Instance
3. Running FireSim Simulations
4. Building Your Own Hardware Designs (FireSim FPGA Images)
ADVANCED DOCS:
Manager Usage (the `firesim` command)
Workloads
Targets
Debugging
Supernode - Multiple Simulated SoCs Per FPGA
Miscellaneous Tips
FireSim Asked Questions
GOLDEN GATE (MIDAS II) DOCS:
Overview & Philosophy
Target Abstraction & Host Decoupling
Target-to-Host Bridges
Bridge Walkthrough
Read the Docs v: latest

that you chose above (one of `us-east-1`, `us-west-2`, `eu-west-1`) and set the default output format to `json`. You will need to generate an AWS access key in the “Security Credentials” menu of your AWS settings (as instructed in <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#access-keys-and-secret-access-keys>).

Again on the `t2.nano` instance, do the following:

```
sudo yum -y install python-pip
sudo pip install boto3
wget https://raw.githubusercontent.com/firesim/firesim/master/scripts/aws-setup.py
python aws-setup.py
```

This will create a VPC named `firesim` and a security group named `firesim` in your account.

2.2.6. Terminate the t2.nano

At this point, we are finished with the general account configuration. You should terminate the t2.nano instance you created, since we do not need it anymore (and it shouldn't contain any important data).

2.2.7. Subscribe to the AWS FPGA Developer AMI

Go to the [AWS Marketplace page for the FPGA Developer AMI](#). Click the button to subscribe to the `FIRESIM FFM` (should be free) and follow the prompts to accept the EULA (but do not launch any instances).

Now, hit next to continue on to setting up our Manager Instance.

Previous

Next

© Copyright 2018, Sagar Karandikar, Howard Mao, Donggyu Kim, David Biancolin, Alon Amid, and Berkeley Architecture Research. Revision 6a4b615d.

Built with [Sphinx](#) using a theme provided by [Read the Docs](#).



Configuring FireSim AWS Infrastructure



Subscribe to the FPGA Developer AMI

The screenshot shows the AWS Marketplace product page for the 'FPGA Developer AMI'. The page includes the AWS logo, a brief description of the AMI, a rating of 4.5 stars from 4 reviews, and a 'Free Tier' badge. A large red box highlights the 'Continue to Subscribe' button. Below the main content, there's a 'Product Overview' section with detailed text about the AMI, followed by a table of metadata and a 'Highlights' box.

FPGA Developer AMI
By: [Amazon Web Services](#) Latest Version: 1.8.0

The FPGA (field programmable gate array) AMI is a supported and maintained CentOS Linux image provided by Amazon Web Services. The AMI is pre-built with FPGA development tools

Linux/Unix ★★★★☆ 4 AWS reviews
Free Tier

Overview **Pricing** **Usage** **Support** **Reviews**

Product Overview

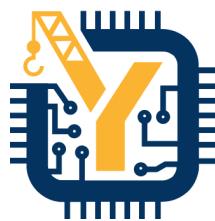
The FPGA (field programmable gate array) AMI is a supported and maintained CentOS Linux image provided by Amazon Web Services. The AMI is pre-built with FPGA development tools and run time tools required to develop and use custom FPGAs for hardware acceleration. The FPGA Developer AMI along with the FPGA Developer Kit(<https://github.com/aws/aws-fpga>) constitutes a development environment which includes scripts and tools for simulating your FPGA design, compiling code, building and registering your AFI (Amazon FPGA Image). Developers can deploy the FPGA developer AMI on an Amazon EC2 instance and quickly provision the resources they need to write and debug FPGA designs in the cloud. The AMI is designed to provide a stable, secure, and high performance development environment. The FPGA AMI is provided at no additional charge to Amazon EC2 users.

Version	1.8.0 Show other versions
By	Amazon Web Services
Categories	High Performance Computing
Operating System	Linux/Unix, CentOS 7.5
Delivery Methods	Amazon Machine Image

Highlights

- Xilinx Vitis 2019.2(v1.8.x), SDx 2019.1(v1.7.x), 2018.3(v1.6.x), 2018.2(v1.5.x) or 2017.4 (v1.4.X) and Free license for F1 FPGA development
- AWS Integration - includes packages and configurations that provide tight integration with Amazon Web Services





Configuring FireSim AWS Infrastructure

The screenshot shows the AWS Marketplace product page for the "FPGA Developer AMI". The top navigation bar includes links for Categories, Delivery Methods, Solutions, Migration Mapping Assistant, Your Saved List, a search bar, and options for Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help.

The main content area displays the "FPGA Developer AMI" logo and title. Below it, there are links for "Product Detail" and "Subscribe". A message states: "To create a subscription, review the pricing information and accept the terms for this software." A "Continue to Configuration" button is visible, along with a note: "You must first review and accept terms."

The "Terms and Conditions" section contains a detailed legal text about the subscription terms, mentioning the End User License Agreement (EULA), AWS Privacy Notice, and AWS Customer Agreement. An "Accept Terms" button is located at the bottom right of this section, which is highlighted with a red box.

The "Amazon Web Services Offer" section provides a table of pricing information for the listed software components. The table has two columns: "FPGA Developer AMI" and "Additional taxes or fees may apply." It lists EC2 Instance Types and their corresponding Software/hr costs:

FPGA Developer AMI	Additional taxes or fees may apply.
EC2 Instance Type	Software/hr
t2.nano	\$0
t2.micro	\$0
t2.small	\$0
t2.medium	\$0
t2.large	\$0
t2.xlarge	\$0





Configuring FireSim AWS Infrastructure

Receive confirmation
for subscribing to the
FPGA Developer AMI

aws marketplace

Categories ▾ Delivery Methods ▾ Solutions ▾ Migration Mapping Assistant Your Saved List

aws FPGA Developer AMI Continue to Configuration

Thank you for subscribing to this product! You can now configure your software. X

< Product Detail Subscribe

Subscribe to this software

You're subscribed to this software. Please see the terms and pricing details below or click the button above to configure your software.

Terms and Conditions

Amazon Web Services Offer

You have subscribed to this software and agreed that your use of this software is subject to the pricing terms and the seller's [End User License Agreement \(EULA\)](#). You agreed that AWS may share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the [AWS Privacy Notice](#). Your use of AWS services remains subject to the [AWS Customer Agreement](#) or other agreement with AWS governing your use of such services.

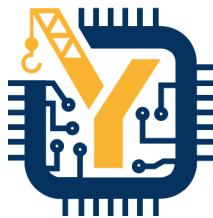
Product	Effective date	Expiration date	Action
FPGA Developer AMI	2/21/2020	N/A	▼ Show Details

[Twitter](#) [AWS Marketplace Blog](#) [RSS Feed](#)

Solutions	Business Applications	Data Products	Sell in AWS Marketplace	AWS Marketplace is hiring!
Data & Analytics	Blockchain	Financial Services Data	Management Portal	Amazon Web Services (AWS) is a dynamic, growing business unit within Amazon.com. We are currently hiring Software Development Engineers, Product Managers, Account Managers, Solutions Architects, Support Engineers, System Engineers, Designers and more. Visit our Careers page or our Developer-specific Careers page to learn more.
DevOps	Collaboration & Productivity	Healthcare & Life Sciences Data	Sign up as a Seller	
Internet of Things	Contact Center	Media & Entertainment Data	Seller Guide	
Infrastructure Software	Content Management	Telecommunications Data	Partner Application	
Machine Learning	CRM	Gaming Data	Partner Success Stories	
Migration	eCommerce		About AWS	
Security				



Lab 3 Instructions - Manager Instance



- Do this when you are starting the second part of the lab
- Note that these instructions diverge from the main FireSim documentation, since we have prepared an image for you with pre-built tools
 - This is in order to save time. This setup process is long enough :)

EE 290-2 Spring 2020

Lab 3: Tiling and Optimization for Accelerators

model implemented in the non-standard version of the Spike ISA simulator. The Spike ISA simulator was originally used as a “golden model” for the RISC-V ISA. As a functional simulator, in Spike every instruction takes only 1 cycle to execute (no matter how complicated the instruction is, or whether it should have memory latency).

A binary of the Spike simulator is located in the software development tools that you get when you source `/home/ff/ee290-2/chipyard-env.sh` in Chipyard. Spike should be on your path after you source this file, so in order to run a software binary in Spike, you should just need to run the following command:

```
spike --extension=gemmini <path/to/software/binary>
```

Spike has various visibility features that may help you while debugging your software implementations. You can read about most of these features in <https://github.com/ucb-bar/esp-isa-sim/#interactive-debug-mode>

You are also provided with a special version of Spike, which includes additional details about the functional simulation of the Gemmini functional model. In order to enable this version of spike, you will need to source the `/home/ff/ee290-2/chipyard-debug-env.sh` file instead of `/home/ff/ee290-2/chipyard-env.sh`. This version generates many print statements, so we recommend using it only when necessary. This version is identical to the regular Spike version (command line options, flags, etc.), with the addition of these print statements.

```
source /home/ff/ee290-2/chipyard-debug-env.sh  
spike --extension=gemmini <path/to/software/binary>
```

3.2 FireSim FPGA-Accelerated Simulation

FireSim is an FPGA-accelerated cycle-exact simulation platform which uses FPGAs on the AWS EC2 public cloud. In contrast to software RTL simulation, FPGA-accelerated simulation enables us to run long workloads (billions-trillions of cycles) within reasonable wall-clock time. Running these workloads in software RTL simulation would take many hours/days/weeks. In contrast to standard FPGA prototyping, FireSim's simulation maintains cycle accurate timing behavior of the entire system (including memory and peripherals). For example, the simulations you are going to run in this lab are going to use a timing-accurate DDR3 memory model. Additional information about FireSim can be found on the FireSim website (<https://firesim.org/>) and in the FireSim documentation at <https://docs.firesim.org/en/latest/>.

FireSim is included as part of the Chipyard framework, which we used in Lab 2. FireSim is located in the `sims/firesim` directory of Chipyard. You will use FireSim in this lab in order to evaluate the performance of real DNN models on the Gemmini accelerator RTL using simulations which take billions of cycles to run.

The remainder of this section will guide you through setting up a FireSim manager instance for this assignment, which you will use for the second half of the assignment. We recommend going through this part of the setup of the manager instance only after you completed the first part of the assignment (on the local `eda` machines), since this will help conserve AWS resources while your instance is not in active use.

FireSim uses a central *manager* instance in order to manage its operations on AWS. In order to set up your FireSim manager instance, head to the [EC2 Management Console](#). In the top right corner, ensure that the correct region is selected.

To launch a manager instance, follow these steps:

1. From the main page of the EC2 Management Console, click **Launch Instance**. We use an on-demand instance here, so that your data is preserved when you stop/start the instance, and your data is not lost when pricing spikes on the spot market.

10

EE 290-2 Spring 2020

Lab 3: Tiling and Optimization for Accelerators

2. When prompted to select an AMI, search for the following AMI name: **Berkeley EE290-2 FireSim Manager AMI – Spring 2020**. If you have completed the form in the the introduction section, and received an AWS promocode from us, it should appear in the “My AMIs” section (we are sharing this AMI manually with you, since there are limitations on public AMIs in AWS. If more than a day has passed since you completed the form in the introduction section and the AMI doesn't appear in your , please email us). ****DO NOT USE ANY OTHER VERSION,**⁷**.

3. When prompted to choose an instance type, select the instance type of your choosing. A good choice is a **r5.large**.

4. On the “Configure Instance Details” page, first make sure that the `firesim` VPC is selected in the drop-down box next to “Network”. Any subnet within the `firesim` VPC is fine. Additionally, check the box for “Protect against accidental termination.” This adds a layer of protection to prevent your manager instance from being terminated by accident. You will need to disable this setting before being able to terminate the instance using usual methods.

5. You can skip the “Add Tags” page.

6. On the “Configure Security Group” page, select the “`firesim`” security group that was automatically created for you earlier.

7. On the review page, click the button to launch your instance. Make sure you select the `firesim` key pair that we setup earlier.

Note: once the instances was launched, your AWS account is charged for its use. You should “stop” your manager instance when you are not using it for more than a few hours—especially at night. This will make sure your allocated AWS credits will suffice for this lab.

FireSim recommends using `mosh` instead of `ssh`, or using `ssh` with a screen/tmux session running on your manager instance to ensure that long-running jobs are not killed by a bad network connection to your manager instance. In either case, `ssh` (or `mosh`) into your instance (e.g. `ssh -i firesim.pem centos@YOUR_INSTANCE_IP`). Now that the manager instance is started, copy the private key that you downloaded from AWS earlier when you set up the infrastructure (`firesim.pem`) to `7firesim.pem` on your manager instance. This step is required to give the manager access to the instances it launches for you.

Go into the `firesim` directory in `chipyard/sims/firesim` on the manager instance, and source the `sourceme-f1-manager.sh` file.

```
$ source sourceme-f1-manager.sh
```

Finally, run the `firesim managerinit` command.

```
$ firesim managerinit
```

This will first prompt you to setup AWS credentials on the instance, which allows the manager to automatically manage build/simulation nodes (these are the same credentials you entered in the infrastructure setup section with the `aws configure` command (choose the `us-east-1` region, and `json` default output format).

Next, it will create initial configuration files we will use. Finally, it will prompt you for an email address, which is used to send email notifications upon FPGA build completion and optionally for workload completion. You can leave this blank if you do not wish to receive any notifications.

⁷This is a version of the Amazon FPGA Developers AMI version 1.6 in which we have pre-installed Chipyard, FireSim, the software toolchain, and other dependancies and configurations found on the `ee290` branch of Chipyard, in order to save you time





Setting Up Your Manager Instance

Select “Instances” on the left menu of the EC2 console.

Then select “Launch Instance” on the top left blue button

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with several sections: Instances (with 'Instances' selected), Images, Elastic Block Store, and Network & Security. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, and 'N. Virginia' dropdown. Below the navigation bar, there's a search bar with the placeholder 'Filter by tags and attributes or search by keyword'. In the center, it says 'You do not have any running instances in this region.' and 'First time using EC2? Check out the [Getting Started Guide](#). Click the Launch Instance button to start your own server.' A large blue 'Launch Instance' button is located at the bottom right of this central area. A red box highlights the blue 'Launch Instance' button in the top navigation bar.





Setting Up Your Manager Instance

Search for “Berkeley EE290-2 FireSim Manager AMI – Spring 2021”

Screenshot of the AWS Lambda Step Functions "Choose AMI" step interface. The search bar at the top contains the query "Berkeley EE290-2 FireSim Manager AMI - Spring 2020". A red box highlights this search bar.

The results list shows several AMI options:

- Amazon Linux 2 AMI (HVM), SSD Volume Type** - ami-0a887e401f7654935 (64-bit x86) / ami-002cc39e7bf021a77 (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type** - ami-0e2ff28bfb72a4e45
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** - ami-0c322300a1dd5dc79 (64-bit x86) / ami-03587fa4048e9eb92 (64-bit Arm)
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type** - ami-0df6cfabfbe4385b7 (64-bit x86) / ami-0e83525f5b2878f0 (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-07ebfd5b3428b6f4d (64-bit x86) / ami-0400a1104d5b9caa1 (64-bit Arm)
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

A message at the bottom of the list says: "Are you launching a database instance? Try Amazon RDS." followed by a "Launch a database using RDS" button.

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use





Setting Up Your Manager Instance

You should see 1 result in “My AMIs”.

If you don’t see a result, please make sure you have filled out the form which allocate AWS credit for you. Otherwise, please email us.

Choose this result.

Step 1: Choose an Amazon Machine Image (AMI)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Cancel and Exit

Berkeley EE290-2 FireSim Manager AMI - Spring 2020

No results were found for "Berkeley EE290-2 FireSim Manager AMI - Spring 2020" in the quick start catalog.
The following results for "Berkeley EE290-2 FireSim Manager AMI - Spring 2020" were found in other catalogs:

- 1 results in My AMIs
My AMIs are AMIs owned by you or shared with you
- 3969 results in AWS Marketplace
AWS Marketplace provides partnered Software that is pre-configured to run on AWS

Quick Start (0)

My AMIs (0)

AWS Marketplace (3969)

Community AMIs (0)

Free tier only ⓘ

No AMIs >





Setting Up Your Manager Instance

You will likely need to clear your filters, since the filters by default filter only AMIs that are owned by you (and this one is an AMI that we shared with you)

The screenshot shows the AWS Management Console with the title "Step 1: Choose an Amazon Machine Image (AMI)". The search bar at the top contains the text "Berkeley EE290-2 FireSim Manager AMI - Spring 2020". Below the search bar, there are tabs for "1. Choose AMI", "2. Choose Instance Type", "3. Configure Instance", "4. Add Storage", "5. Add Tags", "6. Configure Security Group", and "7. Review". On the right side, there is a "Cancel and Exit" button and a "N. Virginia" region selection. The main content area displays a search result for "Berkeley EE290-2 FireSim Manager AMI - Spring 2020". It shows a message: "No results were found for 'Berkeley EE290-2 FireSim Manager AMI - Spring 2020' in the My AMIs catalog with your current filters." A blue link "Clear your filters" is highlighted with a red box. Below this, it says "The following results for 'Berkeley EE290-2 FireSim Manager AMI - Spring 2020' were found in other catalogs:" followed by a bullet point: "• 3969 results in AWS Marketplace". A note below states: "AWS Marketplace provides partnered Software that is pre-configured to run on AWS". On the left, there is a sidebar with filters: "Quick Start (0)", "My AMIs (0)" (which is selected), "AWS Marketplace (3969)", "Community AMIs (0)", "Ownership" (with "Owned by me" checked and "Shared with me" unchecked), "Architecture" (with "32-bit (x86)", "64-bit (x86)", and "64-bit (Arm)" options), and "Root device type" (with "EBS" and "Instance store" options). At the bottom, there are links for "Feedback", "English (US)", and copyright information: "© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use".



Setting Up Your Manager Instance



Select “Berkeley
EE290-2 FireSim
Manager AMI – Spring
2021”

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search: Berkeley EE290-2 FireSim Manager AMI - Spring 2020

Quick Start (0)

My AMIs (1)

AWS Marketplace (3969)

Community AMIs (0)

Ownership

- Owned by me
- Shared with me

Architecture

- 32-bit (x86)
- 64-bit (x86)
- 64-bit (Arm)

Root device type

- EBS
- Instance store

Berkeley EE290-2 FireSim Manager AMI - Spring 2020 - ami-0ba6949143bf02c2d

AMI with pre-built FireSim tools and custom modifications for Lab 3 of UC Berkeley EE290-2 Spring 2020 course, Hardware for Machine Learning

Root device type: ebs Virtualization type: hvm Owner: 643070172799 ENA Enabled: Yes

The following results for "Berkeley EE290-2 FireSim Manager AMI - Spring 2020" were found in other catalogs:

- 3969 results in AWS Marketplace

AWS Marketplace provides partnered Software that is pre-configured to run on AWS

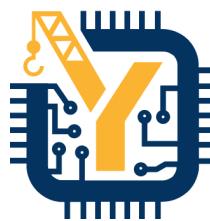
Select

64-bit (x86)

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use





Setting Up Your Manager Instance

Select an r5.large instance type.

We have chosen this instance type as a balance between cost and performance. If you prefer a faster instance that will cost more, you may choose c5.4xlarge (but keep track of your billing!)

Proceed with “Next: Configure Instance Details” (**Not “Review and Launch”**)

Screenshot of the AWS CloudFormation console showing the "Step 2: Choose an Instance Type" page. The r5.large instance type is highlighted with a red box. The "Review and Launch" button is also highlighted with a red box at the bottom right.

Instance Type	Cores	Threads per Core	Memory (GiB)	Storage	Network	Optimized	Up to 10 Gigabit	Yes
r5d.xlarge	4	32	1 x 150 (SSD)	Yes	Up to 10 Gigabit	Yes		
r5d.2xlarge	8	64	1 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes		
r5d.4xlarge	16	128	2 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes		
r5d.8xlarge	32	256	2 x 600 (SSD)	Yes	10 Gigabit	Yes		
r5d.12xlarge	48	384	2 x 900 (SSD)	Yes	10 Gigabit	Yes		
r5d.16xlarge	64	512	4 x 600 (SSD)	Yes	20 Gigabit	Yes		
r5d.24xlarge	96	768	4 x 900 (SSD)	Yes	25 Gigabit	Yes		
r5.large	2	16	EBS only	Yes	Up to 10 Gigabit	Yes		
r5.xlarge	4	32	EBS only	Yes	Up to 10 Gigabit	Yes		
r5.2xlarge	8	64	EBS only	Yes	Up to 10 Gigabit	Yes		
r5.4xlarge	16	128	EBS only	Yes	Up to 10 Gigabit	Yes		
r5.8xlarge	32	256	EBS only	Yes	10 Gigabit	Yes		
r5.12xlarge	48	384	EBS only	Yes	10 Gigabit	Yes		
r5.16xlarge	64	512	EBS only	Yes	20 Gigabit	Yes		
r5.24xlarge	96	768	EBS only	Yes	25 Gigabit	Yes		
r5.metal	96	768	EBS only	Yes	25 Gigabit	Yes		
r4.large	2	15.25	EBS only	Yes	Up to 10 Gigabit	Yes		





Setting Up Your Manager Instance

Select the “firesim” VPC under “Network”

Protect the instance against accidental termination

Proceed with “Next: Add Storage” (Not “Review and Launch”)

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-0f8f5e34d7b478ee9 | firesim Create new VPC

Subnet: subnet-050459788872c5bc us-east-1a Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open Create new Capacity Reservation

IAM role: None Create new IAM role

CPU options: Specify CPU options

Shutdown behavior: Stop

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

EBS-optimized instance: Launch as EBS-optimized instance

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Elastic Inference: Add an Elastic Inference accelerator
Additional charges apply.

Feedback English (US) Cancel Previous Review and Launch Next: Add Storage





Setting Up Your Manager Instance

150 GiB of storage
should be enough

Proceed with “Next:
Add Tags” (**Not**
“Review and
Launch”)

The screenshot shows the AWS EC2 instance setup process at Step 4: Add Storage. The page title is "Step 4: Add Storage". Below it, a note says: "Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2." A table lists the storage configuration:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-05bfcc72e86d17c85	150	General Purpose SSD (gp2)	450 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

A button "Add New Volume" is visible below the table. A note at the bottom says: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions." At the bottom right, buttons for "Cancel", "Previous", "Review and Launch", and "Next: Add Tags" are shown, with "Next: Add Tags" highlighted by a red box.





Setting Up Your Manager Instance

No need to do anything with tags

Proceed with “Next: Configure Security Group” (Not “Review and Launch”)

The screenshot shows the AWS EC2 instance setup wizard at Step 5: Add Tags. The navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags (which is highlighted), Configure Security Group, and Review. A message at the top states: "Step 5: Add Tags. A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources." Below this, there are fields for Key (128 characters maximum) and Value (256 characters maximum). A note says, "This resource currently has no tags. Choose the Add tag button or click to add a Name tag. Make sure your IAM policy includes permissions to create tags." An "Add Tag" button is present, with the note "(Up to 50 tags maximum)". At the bottom, there are buttons for Cancel, Previous, Review and Launch (which is blue and highlighted with a red box), and Next: Configure Security Group.





Setting Up Your Manager Instance

Select an existing security group, and select the firesim security group

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-07571f1322f3b0449	default	default VPC security group	Copy to new
sg-07615283b28979b88	firesim	firesim security group	Copy to new

Proceed with “Review and Launch”

Inbound rules for sg-07615283b28979b88 (Selected security groups: sg-07615283b28979b88)

Type	Protocol	Port Range	Source	Description
Custom UDP Rule	UDP	60000 - 61000	0.0.0.0/0	mosh
Custom UDP Rule	UDP	60000 - 61000	::/0	mosh
SSH	TCP	22	0.0.0.0/0	
Custom TCP Rule	TCP	10000 - 11000	0.0.0.0/0	firesim network mo...
Custom TCP Rule	TCP	10000 - 11000	::/0	firesim network mo...
RDP	TCP	3389	0.0.0.0/0	remote desktop

Cancel Previous **Review and Launch**

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use





Setting Up Your Manager Instance

Review the details of the instance, and click “Launch”

Screenshot of the AWS Step 7: Review Instance Launch page.

The page shows the following steps:

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Berkeley EE290-2 FireSim Manager AMI - Spring 2020 - ami-0ba6949143bf02c2d
AMI with pre-built FireSim tools and custom modifications for Lab 3 of UC Berkeley EE290-2 Spring 2020 course, Hardware for Machine Learning
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
r5.large	10	2	16	EBS only	Yes	Up to 10 Gigabit

Security Groups

Security Group ID	Name	Description
sg-07615283b28979b88	firesim	firesim security group

All selected security groups inbound rules

Type (i)	Protocol (i)	Port Range (i)	Source (i)	Description (i)
Custom UDP Rule	UDP	60000 - 61000	0.0.0.0/0	mosh

Buttons at the bottom right: Cancel, Previous, **Launch** (highlighted with a red box).





Setting Up Your Manager Instance

Choose the “firesim” key pair, which we created in the earlier setup stages.

The screenshot shows the AWS CloudFormation console during the "Step 7: Review Instance Launch" process. The main page displays instance details such as AMI, instance type (r5.large), and security group (firesim). A modal dialog titled "Select an existing key pair or create a new key pair" is overlaid. The dialog contains instructions about key pairs and a dropdown menu where "firesim" is selected. A red box highlights the "Launch Instances" button at the bottom right of the modal. The status bar at the bottom indicates "7. Review".





Setting Up Your Manager Instance

Your instance is now launching.

View it in the instances list

AWS Services Resource Groups N. Virginia Support

Launch Status

Your instances are now launching
The following instance launches have been initiated: i-0a1893d7bdd1fd0f0 [View launch log](#)

Get notified of estimated charges
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

How to connect to your Linux instance	Amazon EC2: User Guide
Learn about AWS Free Usage Tier	Amazon EC2: Discussion Forum

While your instances are launching you can also

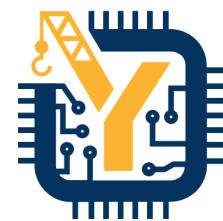
- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

[View Instances](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Setting Up Your Manager Instance

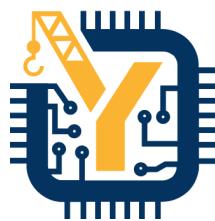


ssh/mosh into your manager instance: ssh -i firesim.pem centos@<INSTANCE IP>

```
The authenticity of host '54.226.188.196 (54.226.188.196)' can't be established.  
ECDSA key fingerprint is SHA256:MEBfVUYG7SYhdWjNReFav8SfqxvKD7tx2TALPbFrpCk.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '54.226.188.196' (ECDSA) to the list of known hosts.  
Last login: Fri Feb 21 18:47:37 2020 from a6.millennium.berkeley.edu
```

```
|__|_ \ /__|/_\ |__\|__\ \ //|/_\|__\|_ _|
|_|_|/_ /(_|/_\|_|_|_|_|\ \ v /|/_\| \ /|_|_|_
|_|_|_| \_/_/_\|_|_|/_| \_/_|/_/_\|_|_|_|_|_
AMI Version: 1.6.0
Readme: /home/centos/src/README.md
GUI Setup Steps: /home/centos/src/GUI_README.md
GUI Setup script: /home/centos/src/scripts/setup_gui.sh
AMI Release Notes: /home/centos/src/RELEASE_NOTES.md
Xilinx Tools: /opt/Xilinx/
Developer Support: https://github.com/aws/aws-fpga/blob/master/README.md#developer-support
Centos Common code: /srv/git/centos-git-common
centos@ip-192-168-3-54.ec2.internal:~$
```





Setting Up Your Manager Instance

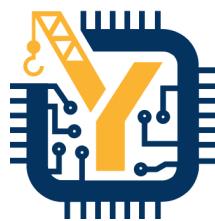
Copy the `firesim.pem` file from your local machine into the home directory of your manager instance

```
centos@ip-192-168-3-54.ec2.internal:~$ scp <LOCAL_MACHINE>/firesim.pem ~/
```

Source the `env.sh` and `sourceme-f1-manager.sh` files in `chipyard` and `chipyard/sims/firesim`. You will need to do this every time you open a terminal session in your manager instance.

```
centos@ip-192-168-3-54.ec2.internal:~$ cd chipyard/sims/firesim
centos@ip-192-168-3-54.ec2.internal:~/chipyard/sims/firesim$ source sourceme-f1-manager.sh
Identity added: /home/centos/firesim.pem (/home/centos/firesim.pem)
success: firesim.pem added to ssh-agent
```





Setting Up Your Manager Instance

Run `firesim managerinit` to initialize your manager. This will require similar details to the `aws configure` command from earlier

```
centos@ip-192-168-3-54.ec2.internal:~/chipyard/sims/firesim$ firesim managerinit
FireSim Manager. Docs: http://docs.firesim.im
Running: managerinit
Running aws configure. You must specify your AWS account info here to use the FireSim Manager.
[localhost] local: aws configure
AWS Access Key ID [None]: XXXXXXXXXXXX
AWS Secret Access Key [None]: XXXXXXXXXXXX
Default region name [None]: us-east-1
Default output format [None]: json
Backing up initial config files, if they exist.
Creating initial config files from examples.
If you are a new user, supply your email address [abc@xyz.abc] for email notifications (leave blank if you do
not want email notifications): john.doe@berkeley.edu
You should receive a message at
john.doe@berkeley.edu
asking to confirm your subscription to FireSim SNS Notifications. You will not
receive any notifications until you click the confirmation link.
FireSim Manager setup completed.
```





Setting Up Your Manager Instance

We're done with setting up the FireSim manager instance!

You can now either:

- Proceed to execute the second part of the lab assignment (evaluation of complete DNNs on Gemmini in FirSim)
- Stop your manager instance, until you are ready for the second part of the assignment





Stopping Your Manager Instance

Stopping your manager instance:

Right click on the instance entry in the table, and within the “Instance State” menu, select “Stop”.

A stopped instance is like “sleep” on your laptop. While the instance is stopped (as opposed to terminated), you are still charged a certain amount of money, but significantly less.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'New EC2 Experience', 'Events', 'Tags', 'Reports', and sections for 'INSTANCES', 'IMAGES', 'ELASTIC BLOCK STORE', 'NETWORK & SECURITY', and 'KEY PAIRS'. The main area displays a table of instances. One instance, 'i-0a1893d7bdd1fd0f0', is selected. A context menu is open over this instance, with 'Instance State' expanded. The 'Stop' option is highlighted with a red box. Below the table, there's a detailed view for the selected instance, including fields for 'Description', 'Status Checks', 'Monitoring', 'Tags', and 'Usage Instructions'. The 'Description' tab is active, showing details like Instance ID, Instance state, Instance type, and VPC ID. The 'Status Checks' tab indicates 2/2 checks passed. The 'Tags' tab shows a single tag 'firesim'. The 'Usage Instructions' tab provides information about Compute Optimizer and scheduled events. At the bottom, there are links for 'Feedback', 'English (US)', and copyright information: '© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.' followed by 'Privacy Policy' and 'Terms of Use'.



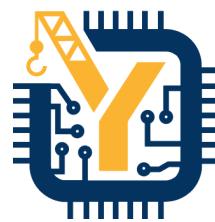
Billing



- We really don't want you to get charged any (real) money
- Check your billing!
- Billing in AWS is not immediate. It is delayed by several hours/days after your actual usage
- If you're running out of credits, please contact the course staff



Billing



In the user menu on the top right corner of your EC2 console, select “My Billing Dashboard”

The screenshot shows the AWS EC2 console interface. On the left is a navigation sidebar with sections like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances, with one row selected for viewing details. At the top right, there is a user menu with options such as My Account, My Organization, My Billing Dashboard (which is highlighted with a red box), Orders and Invoices, My Security Credentials, and Sign Out. Below the main table, a detailed view of the selected instance (i-0a1893d7bdd1fd0f0) is shown, including its description, status checks, monitoring, tags, and usage instructions.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
i-0a1893d7bdd1fd0f0	r5.large	us-east-1d	stopped	None			

Instance: i-0a1893d7bdd1fd0f0 Private IP: 192.168.3.54

Description

Instance ID: i-0a1893d7bdd1fd0f0
Instance state: stopped
Instance type: r5.large
Finding: Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS: ip-192-168-3-54.ec2.internal
Private IPs: 192.168.3.54
Secondary private IPs:
VPC ID: vpc-0f8f5e34d7b478ee9 (firesim)

Public DNS (IPv4): -
IPv4 Public IP: -
IPv6 IPs: -
Elastic IPs: -
Availability zone: us-east-1d
Security groups: firesim, view inbound rules, view outbound rules
Scheduled events: -
AMI ID: Berkeley EE290-2 FireSim Manager AMI - Spring 2020 (ami-0ba6949143bf02c2d)

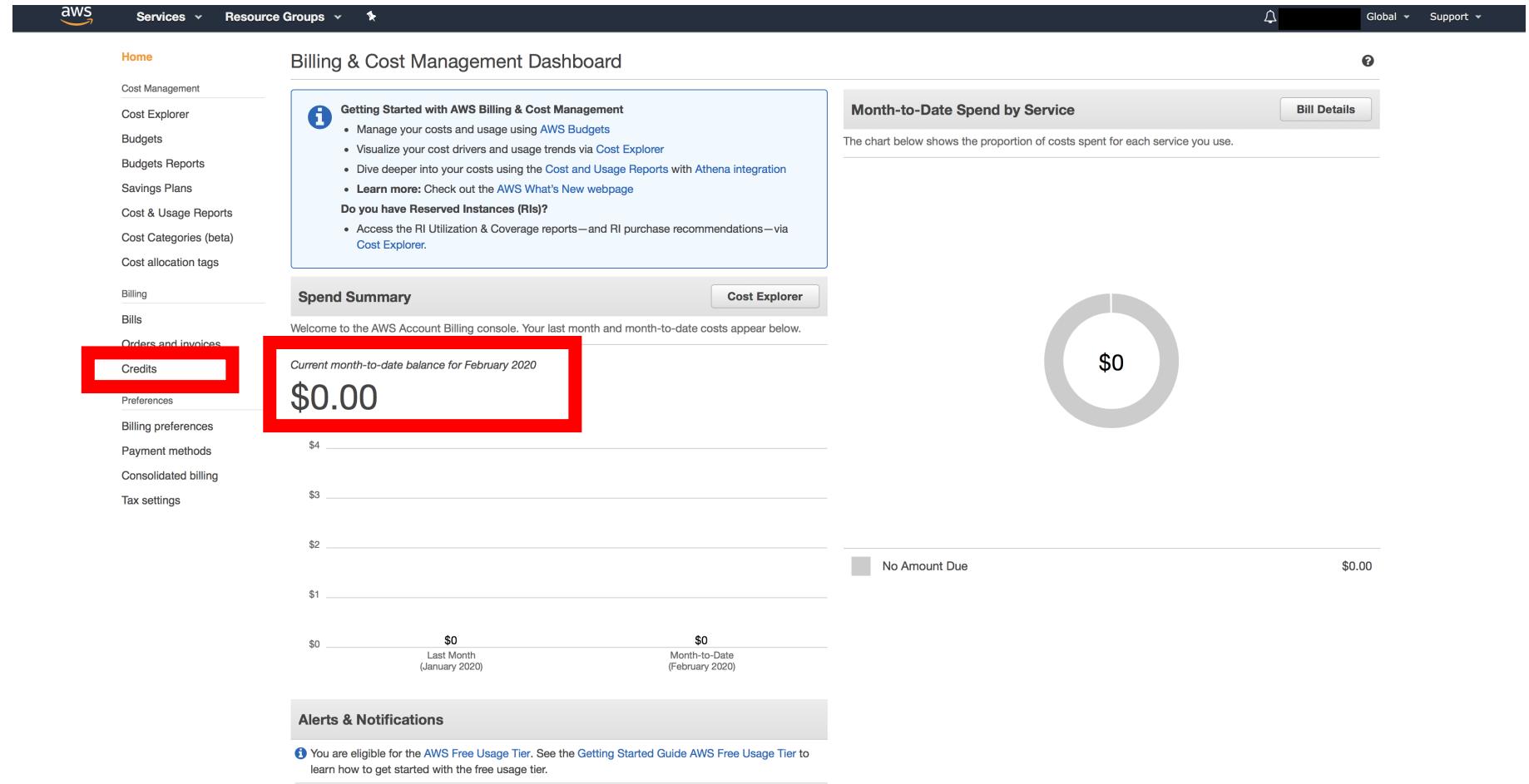


Billing



Spending summary: make sure this doesn't exceed \$150 (because this will mean you're getting very close to your \$200 credit limit)

You may also view your promo code credit usage in the “Credit” section of the menu of the left



Billing



You will see your class credit entry in the table, and the amount used out of it.

If you are running out of credits, please contact the course staff. We have some extra promo codes for special cases.

Screenshot of the AWS Billing Credits page:

The page shows a sidebar menu with the following items under the "Billing" section:

- Home
- Cost Management
- Cost Explorer
- Budgets
- Budgets Reports
- Savings Plans
- Cost & Usage Reports
- Cost Categories (beta)
- Cost allocation tags
- Credits** (highlighted in orange)
- Preferences
- Billing preferences
- Payment methods
- Consolidated billing
- Tax settings

The main content area is titled "Credits" and contains the following steps:

- Please enter your code below to redeem your credits.
- Promo Code:
- Security Check: [Refresh Image](#)
- Please type the characters as shown above:
- By clicking "Redeem" you indicate that you have read and agree to the terms of the AWS Promotional Credit Terms & Conditions located [here](#).
- Redeem** button

The table below displays all AWS credits redeemed by your account. Credits are automatically applied to charges associated with qualifying AWS service usage. Please note that the values for used and remaining credit amounts are updated each month when your invoice is finalized.

Expiration Date	Credit Name	Amount Used	Amount Remaining	Applicable Products
01/31/2022	EDU_ENG_FY2020_IC_Q1_1_BERKELEY_400USD	\$0.00	\$400.00	See complete list

Total Credit Amount Remaining (as of 02/01/2020): \$400.00

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



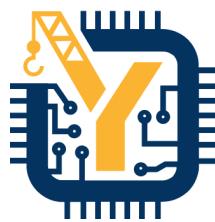
AWS Educate



- Another method to get more credits, is through AWS Educate
- Can get you an extra \$100 credits
- https://aws.amazon.com/education/aws_educate/
- Make sure NOT to choose the Starter Kit (otherwise, you won't have access F1 instances or the promo codes)

The screenshot shows a web browser window for the AWS Educate website at aws.amazon.com/education/awseducate/. The page has a blue header with the AWS logo and navigation links for Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, and Explore More. A prominent orange button says "Join AWS Educate". Below the header, a large banner reads "aws educate" and "Teach Tomorrow's Cloud Workforce Today". A sub-banner below it states: "With the increasing demand for cloud employees, AWS Educate provides an academic gateway for the next generation of IT and cloud professionals. AWS Educate is Amazon's global initiative to provide students and educators with the resources needed to accelerate cloud-related learning." At the bottom, there are three icons representing education: a graduation cap, an open book, and a diploma.





AWS Recap

- Opening an AWS account
- EE290-2 AWS details form
- FireSim infrastructure setup
- FireSim manager instance
- Billing tracking

