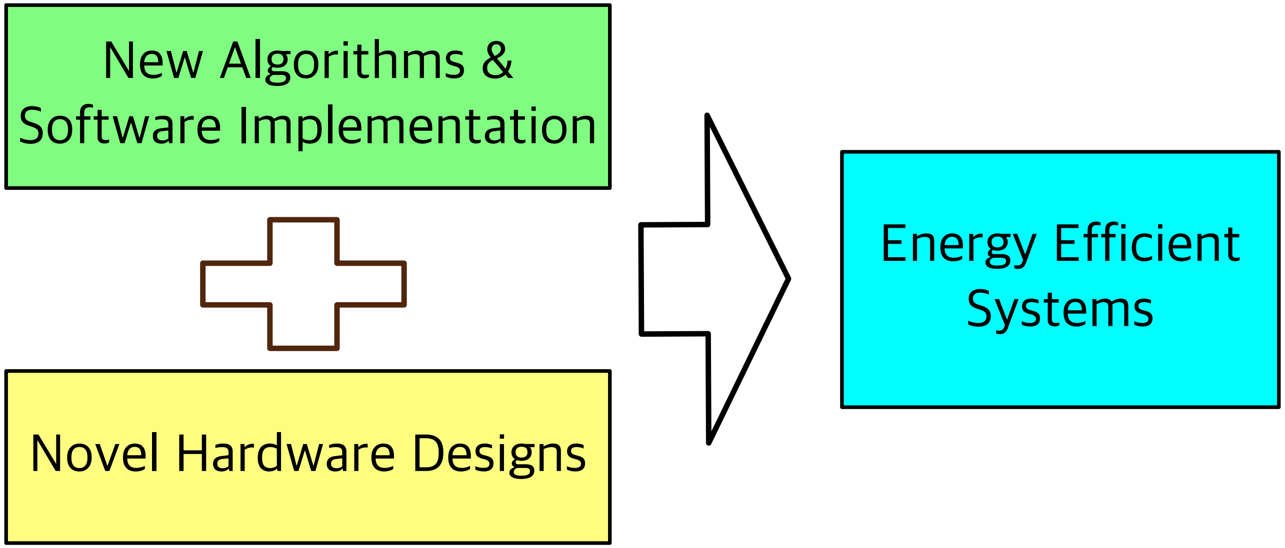
Strober: Fast and Accurate Sample-Based Energy Simulation Framework for Arbitrary RTL

Donggyu Kim, Adam Izraelevitz, Christopher Celio,
Hokeun Kim, Brian Zimmer, Yunsup Lee,
Jonathan Bachrach, Krste Asanović

ISCA 2016
6/20/2016

Energy Evaluation Is Important

- Energy efficiency: key design metric
- Limited energy efficiency improvement from technology scaling
- Great opportunity for
  computer architects & software engineers

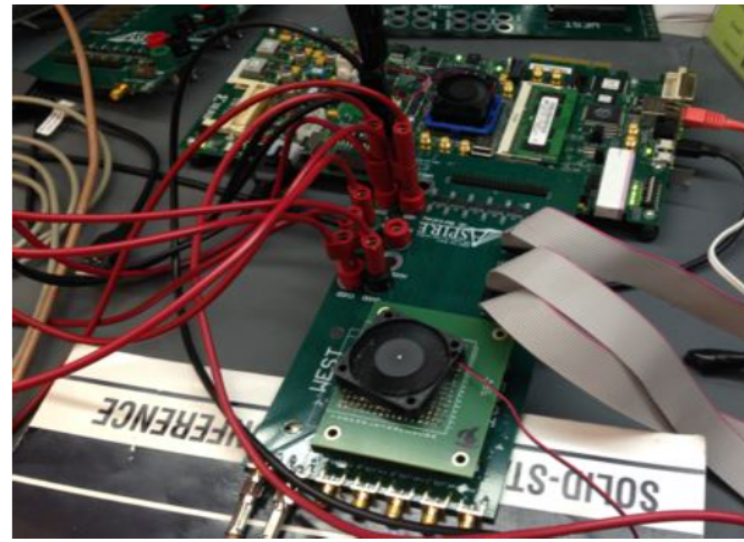


- How to evaluate energy efficiency?

Evaluating Existing Systems Is Easy

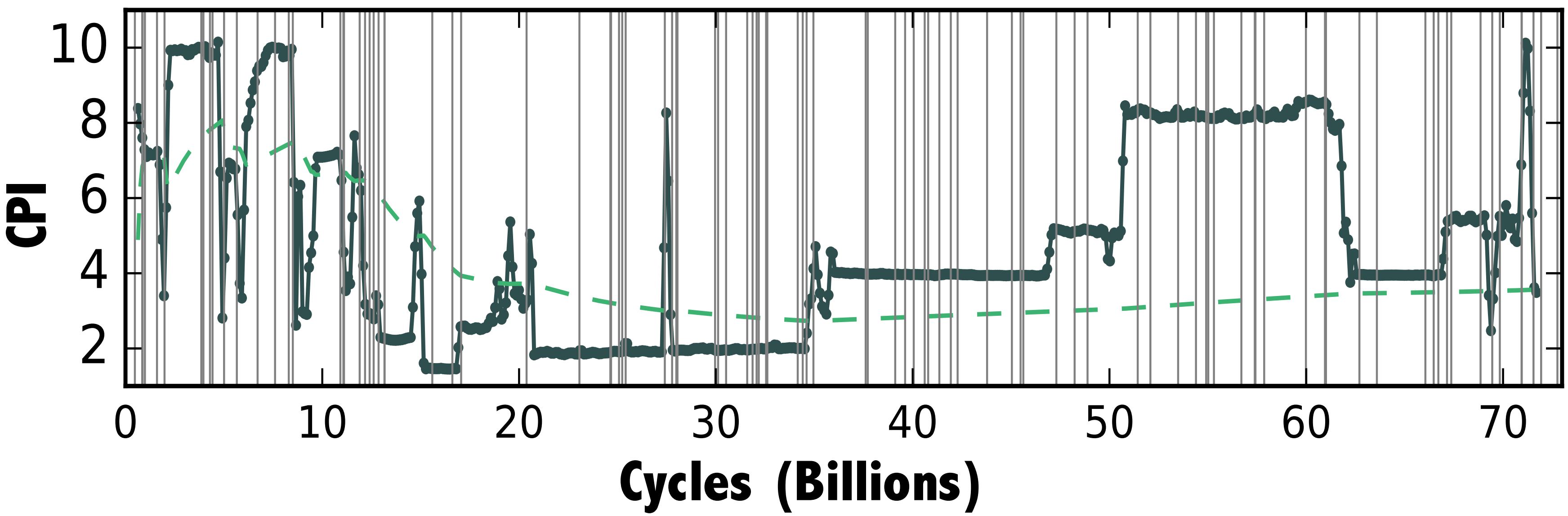
- **Silicon Prototyping**
  - Just run & measure!
  - Very accurate & fast
  - High-latency turn-around
- **What about new systems?**
  - Modeling
  - Simulation

Prototypes from UC Berkeley



What Is A Good Simulation Methodology?

- Fast enough to ***simulate the whole execution of real applications***
  - First 70B cycles of gcc on the in-order-processor (Rocket)



- General, easy to use ***for any hardware design***
  - e.g. Accelerators for deep learning
- ***Minimal modeling errors***

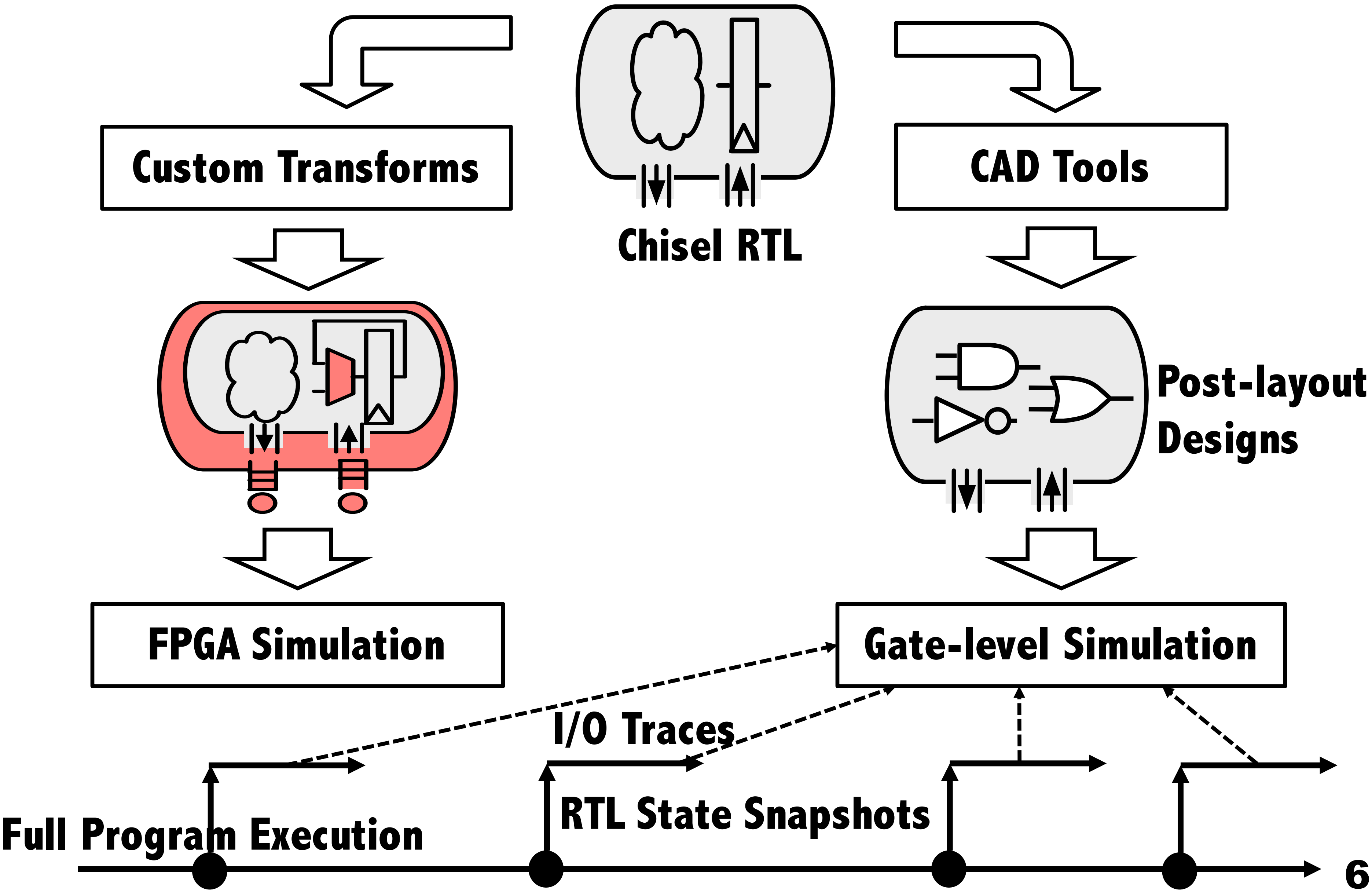
➔ ***Accurate energy prediction for real systems***

Evaluating Novel Systems Is Hard

	Analytic Power Modeling + µarch Software Simulation	RTL / Gate-level Simulation
e.g.	McPAT, Wattch + GEM5	Synopsys VCS
Accuracy	? [1] (Validation against real systems)	✔
Generality	✗ (Additional development efforts)	✔
Speed	✗ (< 300 KHz)	✗ (< 1 KHz)

[1] Xi et. al. "**Quantifying sources of error in McPAT and potential impacts on architectural studies**", HPCA `15

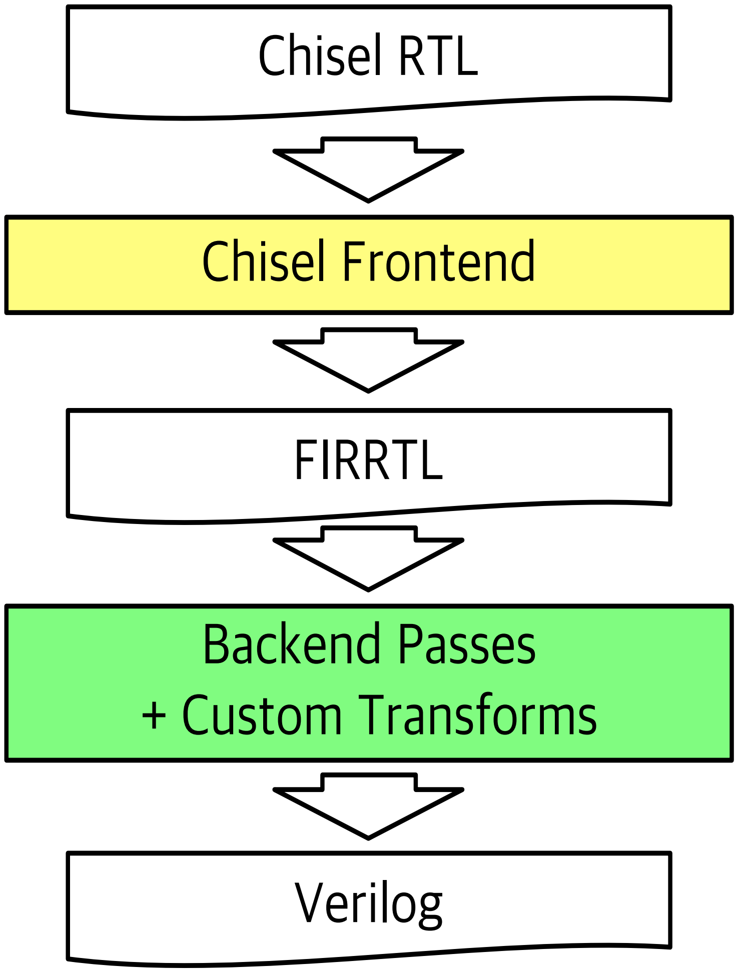
The Strober Framework



Chisel: Constructing Hardware In a Scala Embedded Language

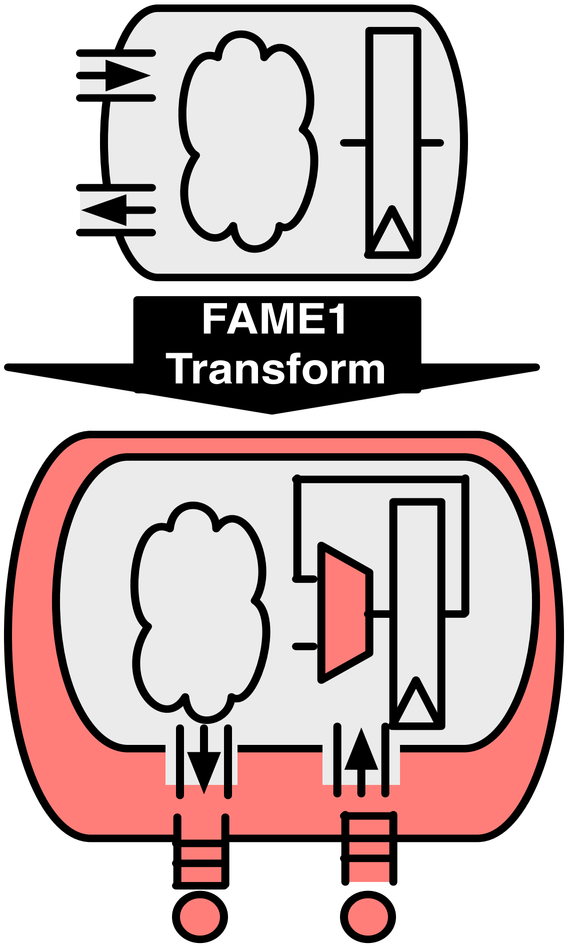
chisel.eecs.berkeley.edu

CHISEL



- Construct hardware generator
  - Clean simple set of design construction primitives
  - Advanced parameterization systems
  - Object-oriented / functional programing with Scala
- Flexible Intermediate Representation for RTL (FIRRTL)
  - ***Custom transforms*** for hardware designs (e.g. scan chains)

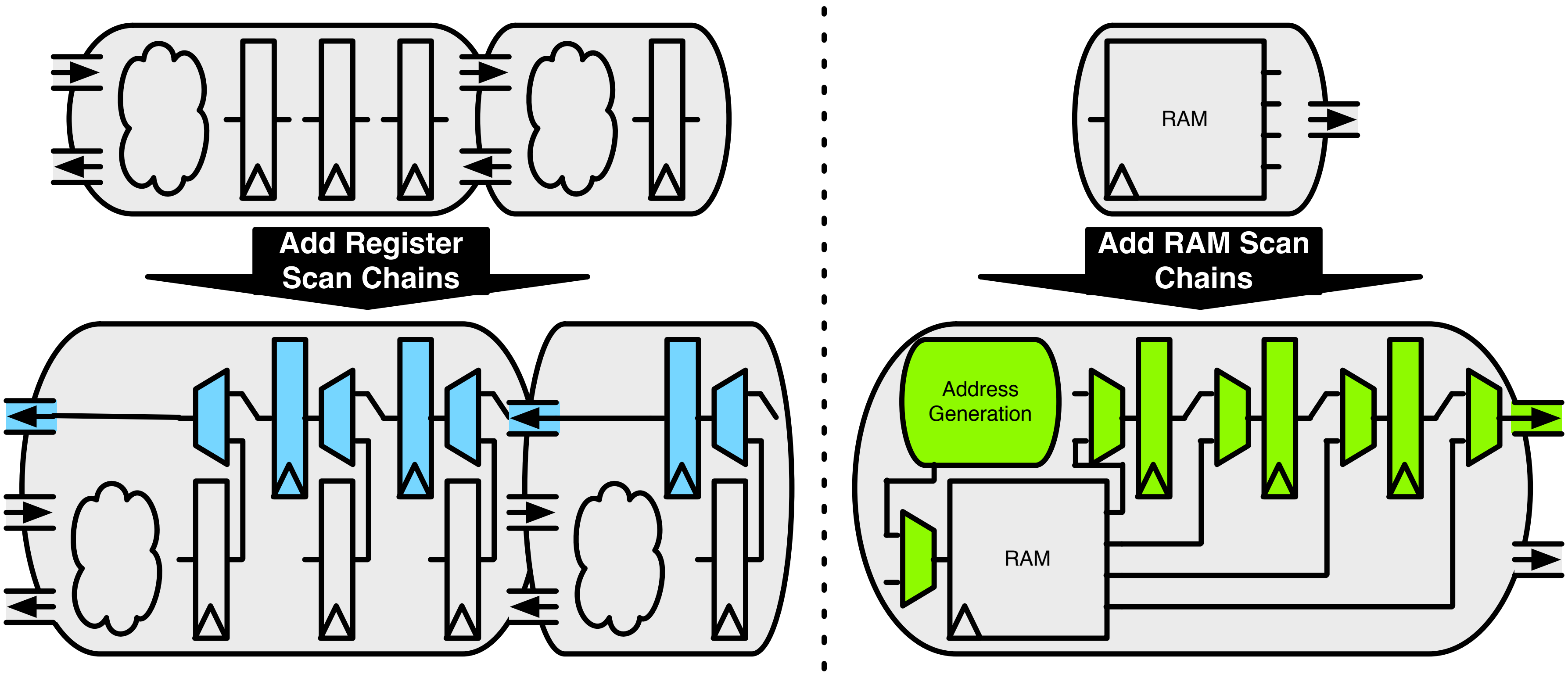
Auto FAME1 Transform



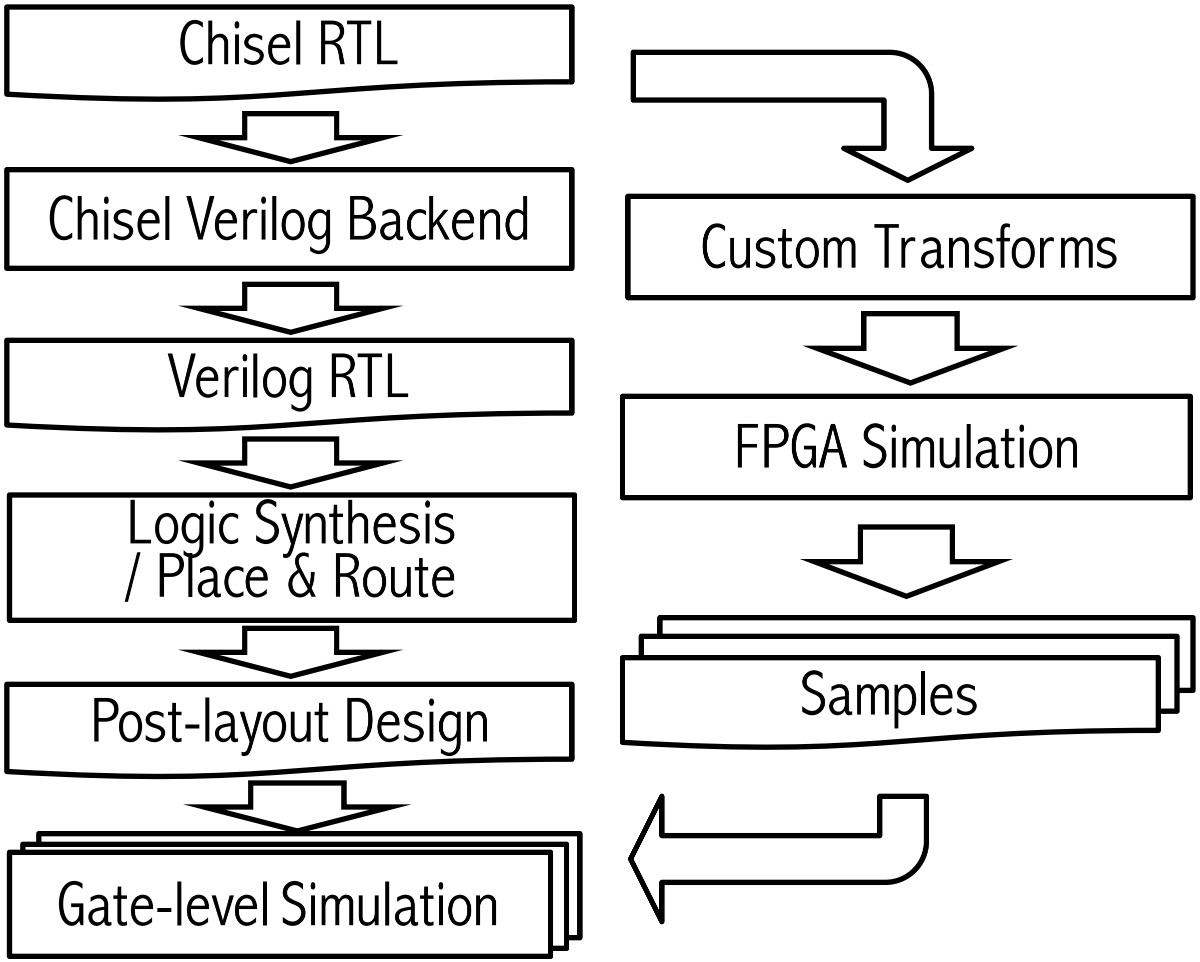
- Problems
  - Performance modeling on heterogeneous FPGA platform
  - Simulation stall to read RTL state snapshots from FPGA simulation
- Solutions: FAME1 Transform
  - ***Automatically*** generates FPGA-accelerated simulators
  - ***Systematically*** attaches enables for registers to stall simulation
  - ***Channels:*** tokens + I/O traces

RTL State Snapshotting

- Add scan chains ***systematically*** & ***automatically*** for
  - **Registers**: values are copied right after the simulation stalls
  - **SRAM/BRAM**: addresses are generated for the read port



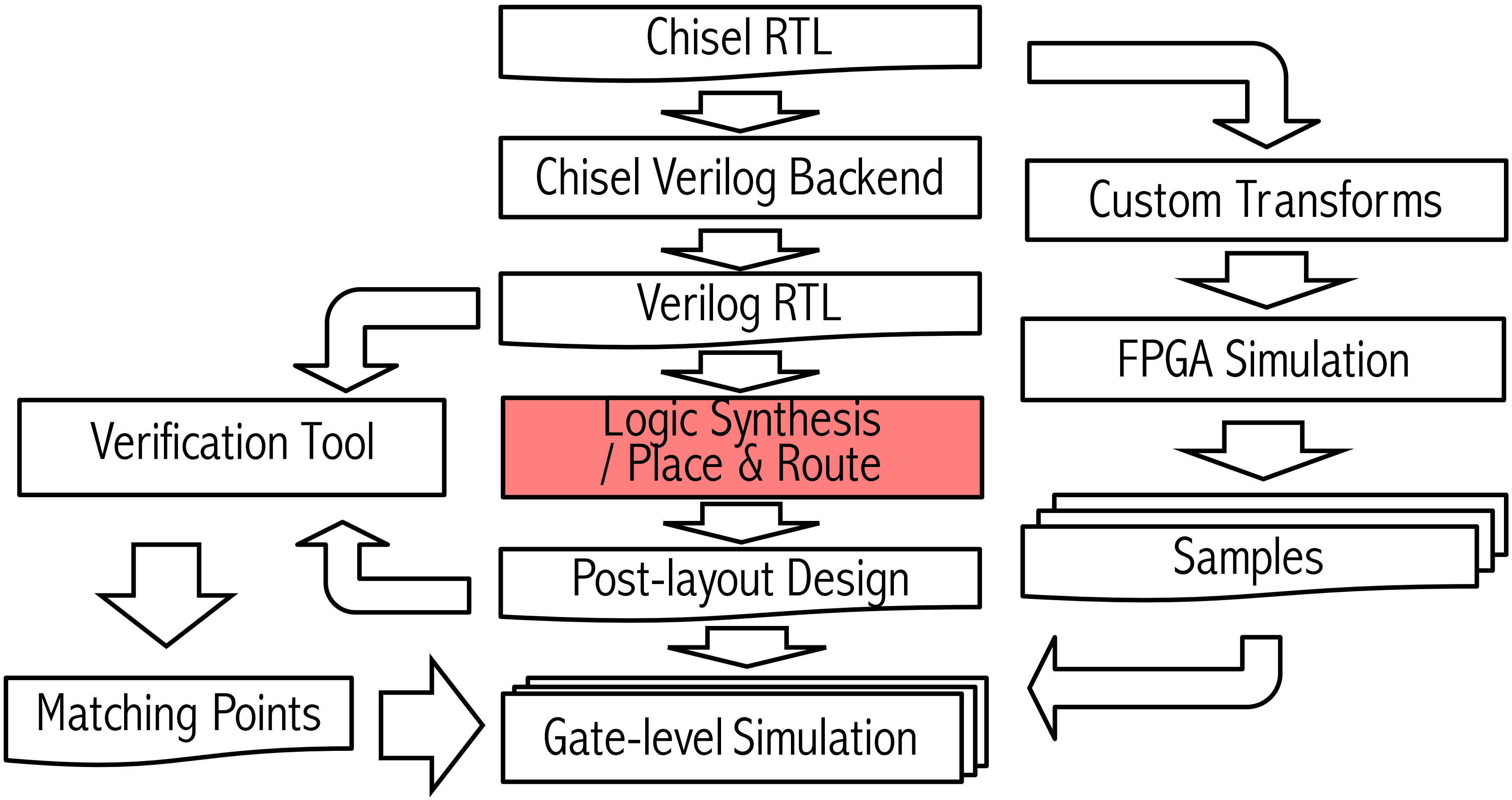
Sample Replays on Gate-level Simulation



- ***Independent RTL State Snapshots***
- ***Parallelize*** sample replays on ***multiple instances*** of gate-level simulation

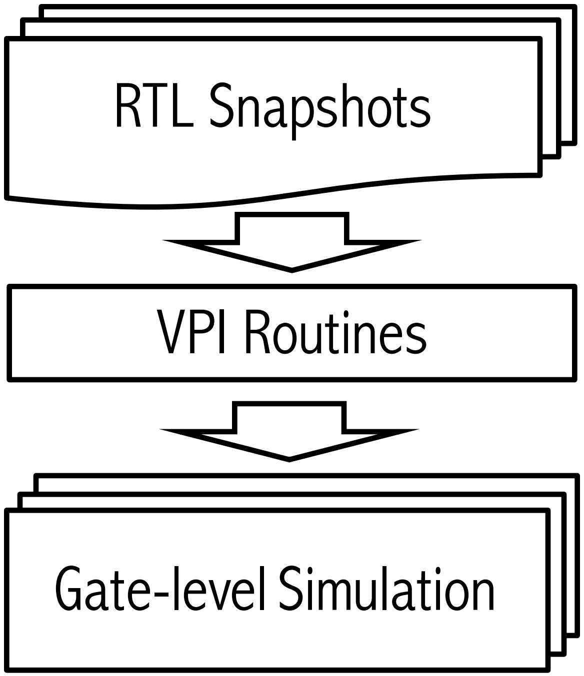
What Is Formal Verification Tool For?

- Problem: RTL name mangling in logic synthesis
- Solution: Formal verification tool
  - ***Find matching points between RTL & gate-level***



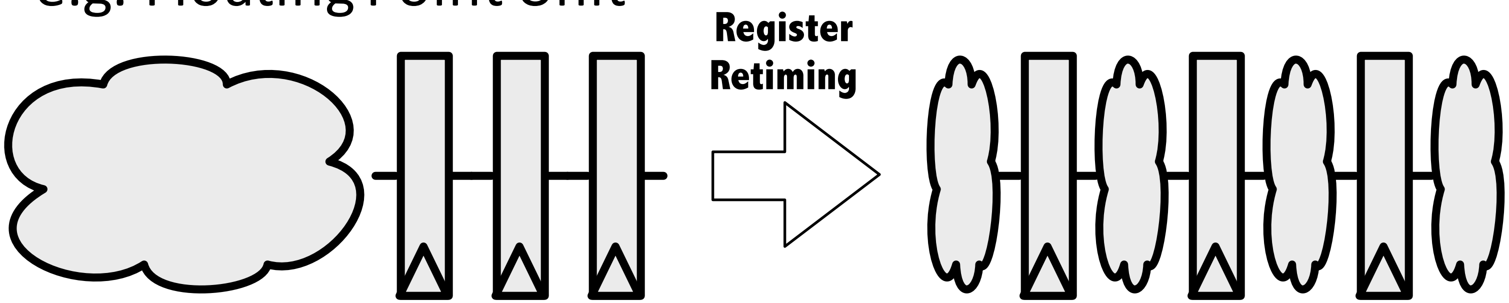
RTL State Loading on Gate-level Simulation

- Problem: Slow RTL state loading on gate-level designs with simulation scripts (e.g. Synopsys UCLI scripts)
  – 400 commands / sec → ***80 sec*** for 35k flip-flops
- Solution: Custom ***VPI routines*** to load RTL state
  – 20,000 commands / sec → ***1.8 sec*** for 35k flip-flops

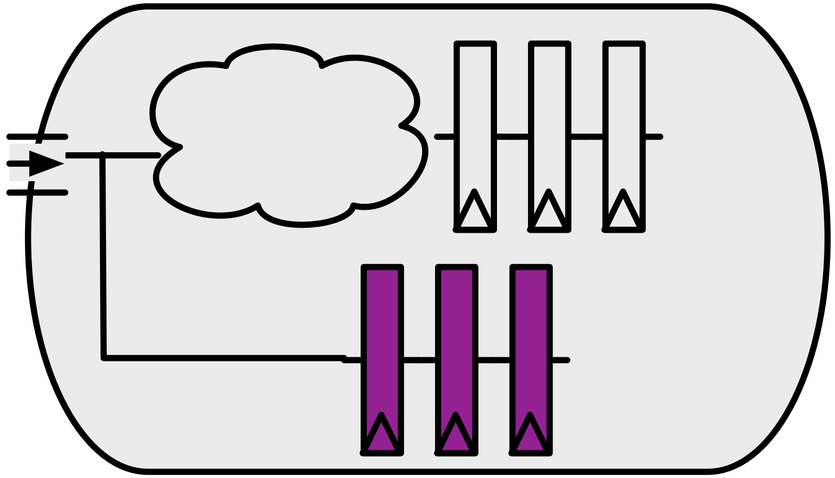


Register Retiming

- Problem: ***Register Retiming*** for a n-cycle datapath
  e.g. Floating Point Unit

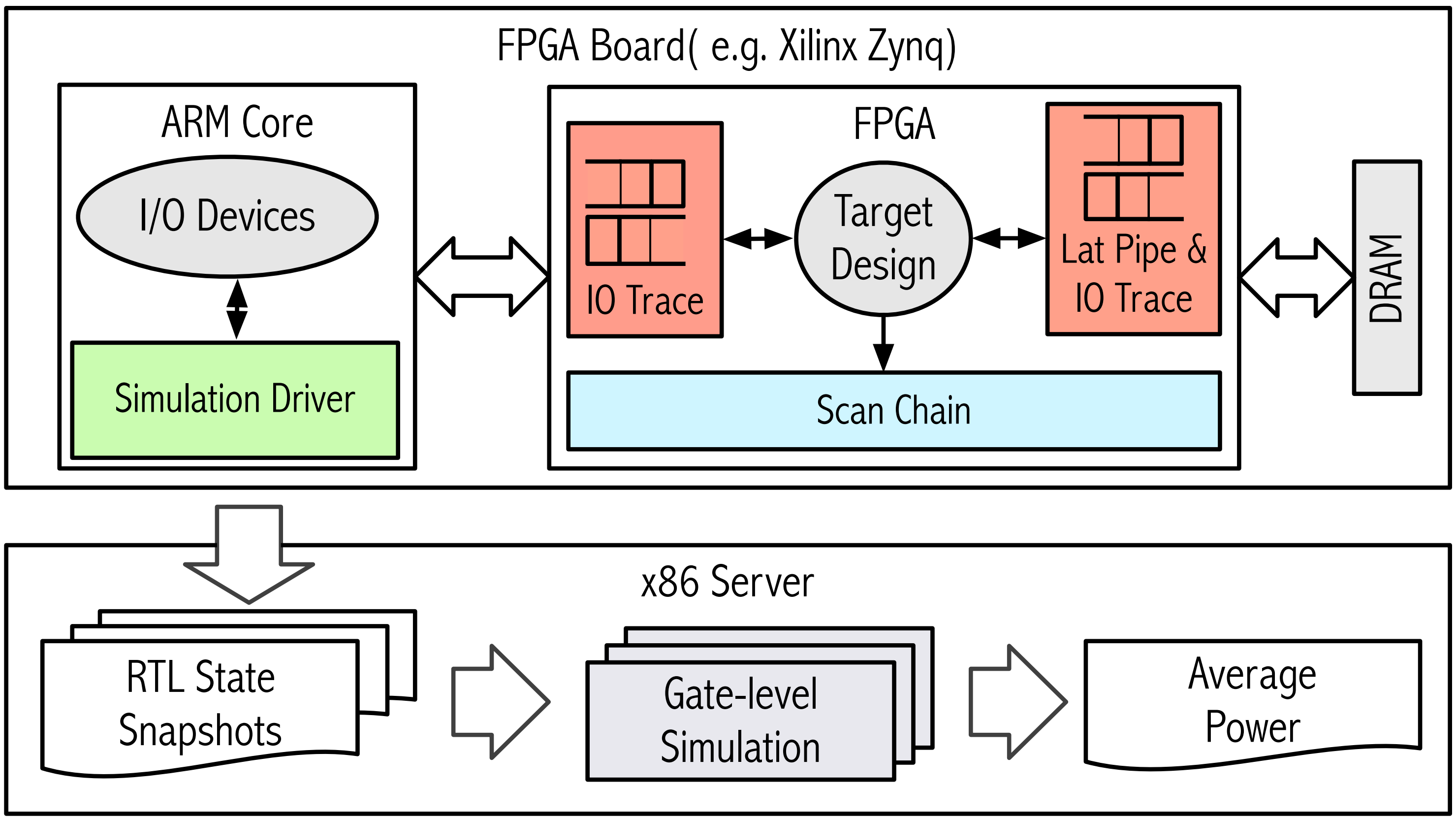


- Solution
  - ***I/O traces for the last n cycles*** in FPGA simulation
  - Forced before replays in gate-level simulation



Simulation Execution Model

- Simulation Driver
  - Periodically reads RTL snapshots from the FPGA Simulator
- ***Automatically*** generate the interface for specific FPGA platforms(e.g. Xilinx Zynq)



How Are We Different?

- Architectural State Sampling (e.g. SMARTS[1])
  - Take architectural state snapshots from software / FPGA functional simulators (e.g. Simics, QEMU, ProtoFlex)
  - Replay snapshots on µarch simulators
  - µarch state warming problems
- FPGA-accelerated Power Estimation (e.g. PrEsto[2])
  - ***Manually*** selects important signals to train power models
  - ***Manually*** adds the power models to the FPGA simulators
  - Needs designers' intuition & additional manual efforts

[1] Wunderlich et. al. "**SMARTS: accelerating microarchitecture simulation via rigorous statistical sampling**", ISCA `02

[2] Sunwoo et. al. "**PrEsto: An FPGA- accelerated Power Estimation Methodology for Complex Systems**", FPGA `10

Target Design: Rocket Chip Generator

https://github.com/ucb-bar/rocket-chip.git

	Rocket [1] (In-order Processor)	BOOM-1w [2] (Out-of-order Processor)	BOOM-2w [2] (Out-of-order Processor)
Fetch-width	1	1	2
Issue-width	1	1	2
Issue slots		12	16
ROB size		24	32
Ld/St entries		8/8	8/8
Physical registers	32(int)/32(fp)	100	110
L1 I$ / D$	16 KiB / 16 KiB	16 KiB / 16 KiB	16 KiB / 16 KiB
DRAM latency	100 cycles	100 cycles	100 cycles

[1] Asanović et. al. "**The Rocket Chip Generator**", UCB Tech Report
[2] Celio et. al. "**The Berkeley Out-of-Order Machine(BOOM): An Industry-Competitive, Synthesizable, Parameterized RISC-V Processor**", UCB Tech Report

How Fast Is Strober?

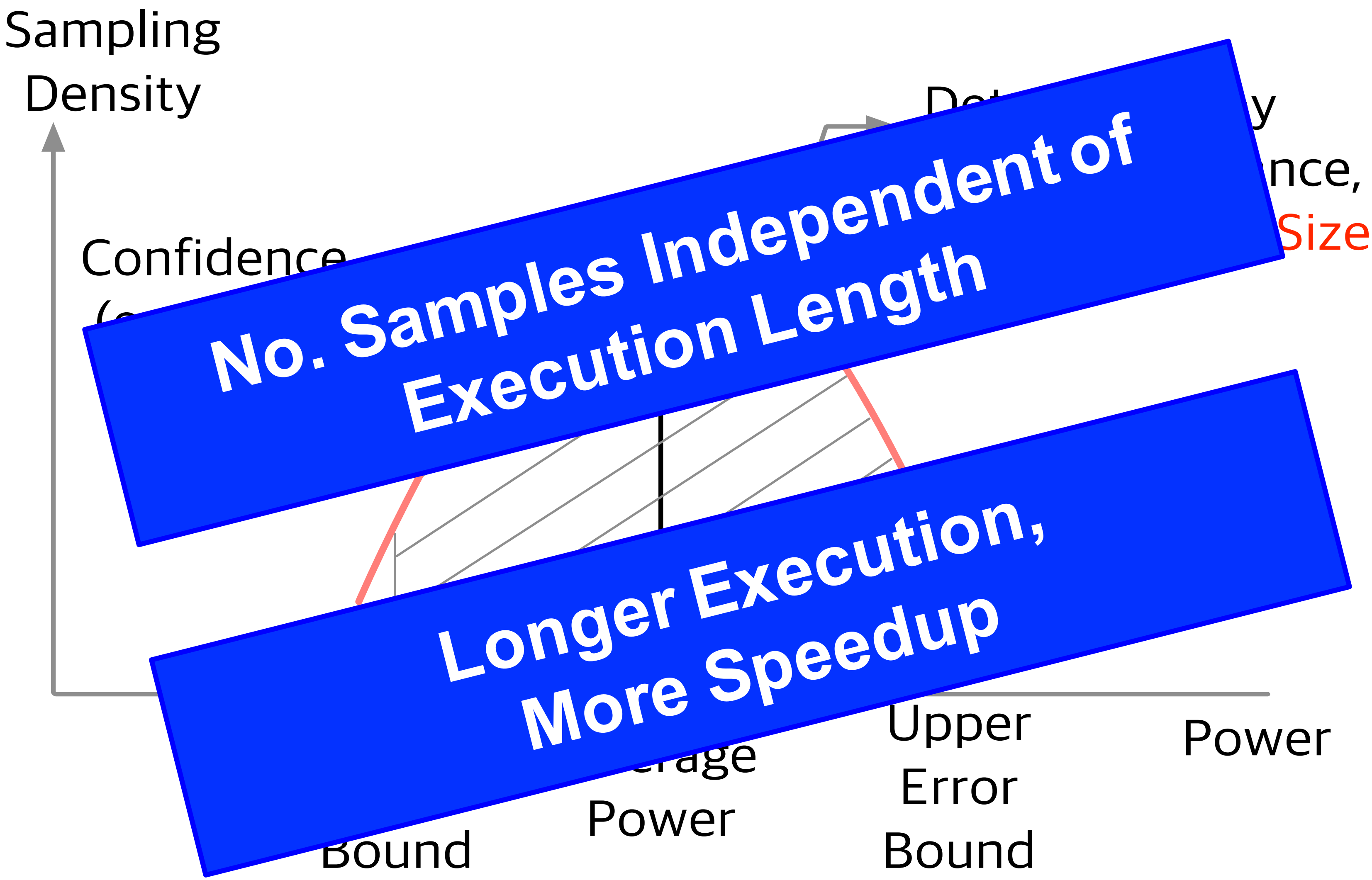
- Example: Two-way Out-of-order Processor(BOOM)

Execution Length	100B cycles	FPGA Synthesis Time	1 hour
# of snapshots	100	FPGA Clock Rate	50 MHz
Replay Length	1000 cycles	FPGA Simulation Speed	3.6 MHz
# of Instances of Gate-level Simulation	10	Gate-level Simulation Speed	12Hz

  – ASIC tool chain can be parallelized with FPGA Simulation

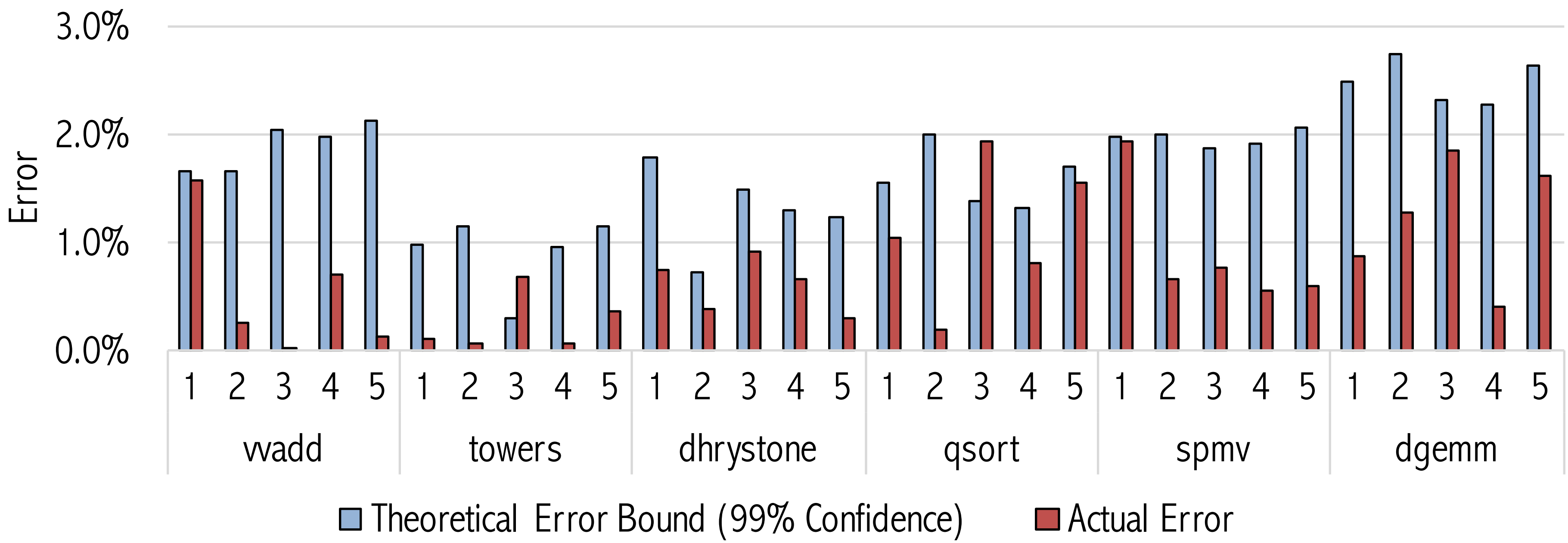
- Simulation Performance Comparisons
  – Gate-level simulation = **264 years** (100B cycles / 12 Hz)
  – ***Fastest*** μarch software simulation = **3.86 days** (100B cycles / 300 KHz)
  – FPGA simulation = **7.7 hours** (100B cycles / 3.6 MHz)
  – Sample Recording = **1.0 hour** (1.3 x 100 x 2 ln(10B / (100 x 1000)))
  – Sample Replays = **39 min** (100 ✕ (1000 cycles / 12 Hz + 2.5 min) / 10)
  – Strober Total = **9.4 hours**

Underlying Theory: Central Limit Theorem



How Accurate Is Strober?

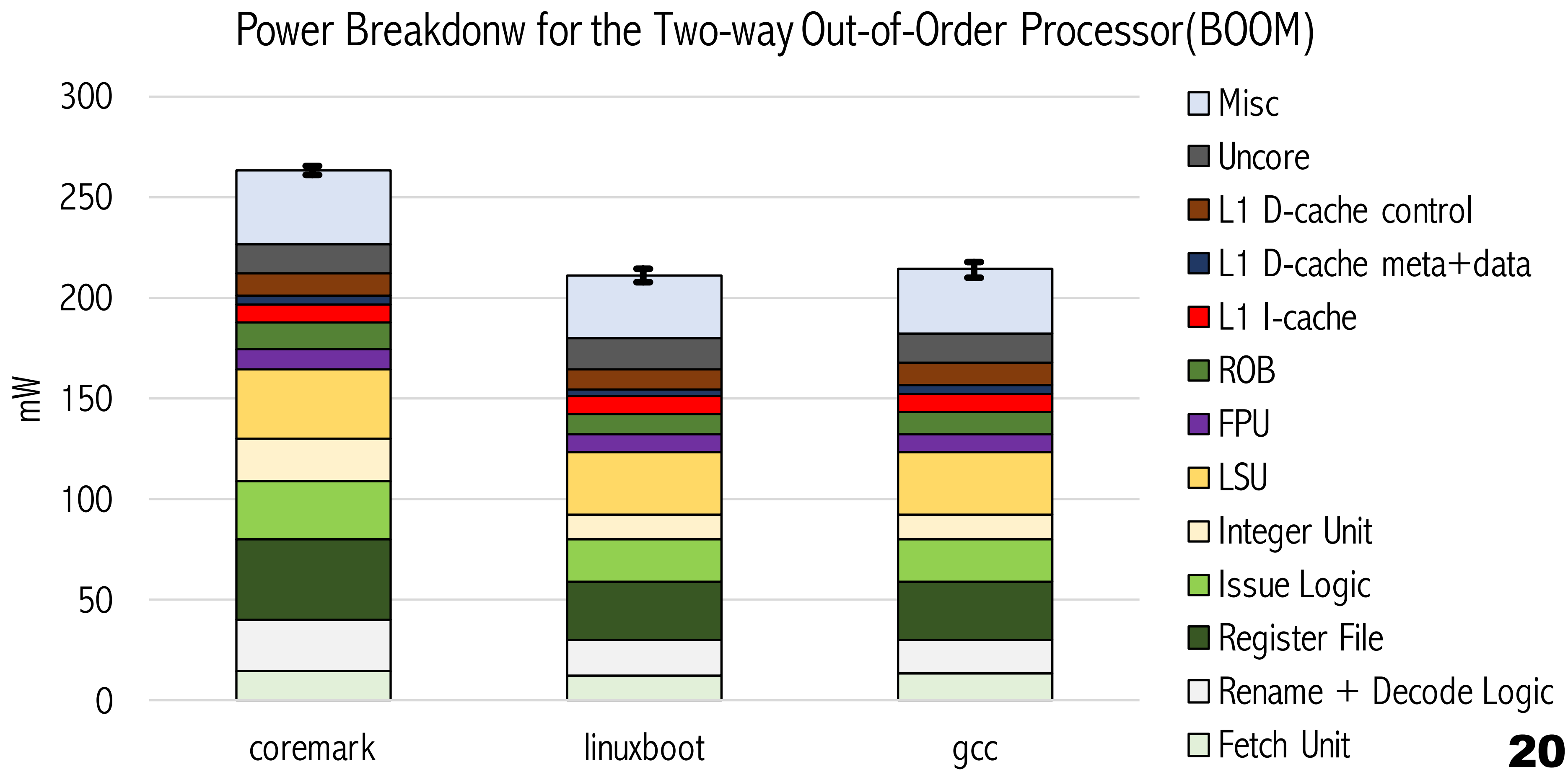
- Technology: TSMC 45nm
- Theoretical Error Bounds vs. Actual Errors



  - Replay 30 random sample snapshots of 128 cycles
  - Repeat five times for benchmarks on the in-order-processor(Rocket)
  - Theoretical error bound: computed from Central Limit Theorem
  - Actual error: compared against the whole execution of benchmarks
  - ***Errors are independent of the length of execution***

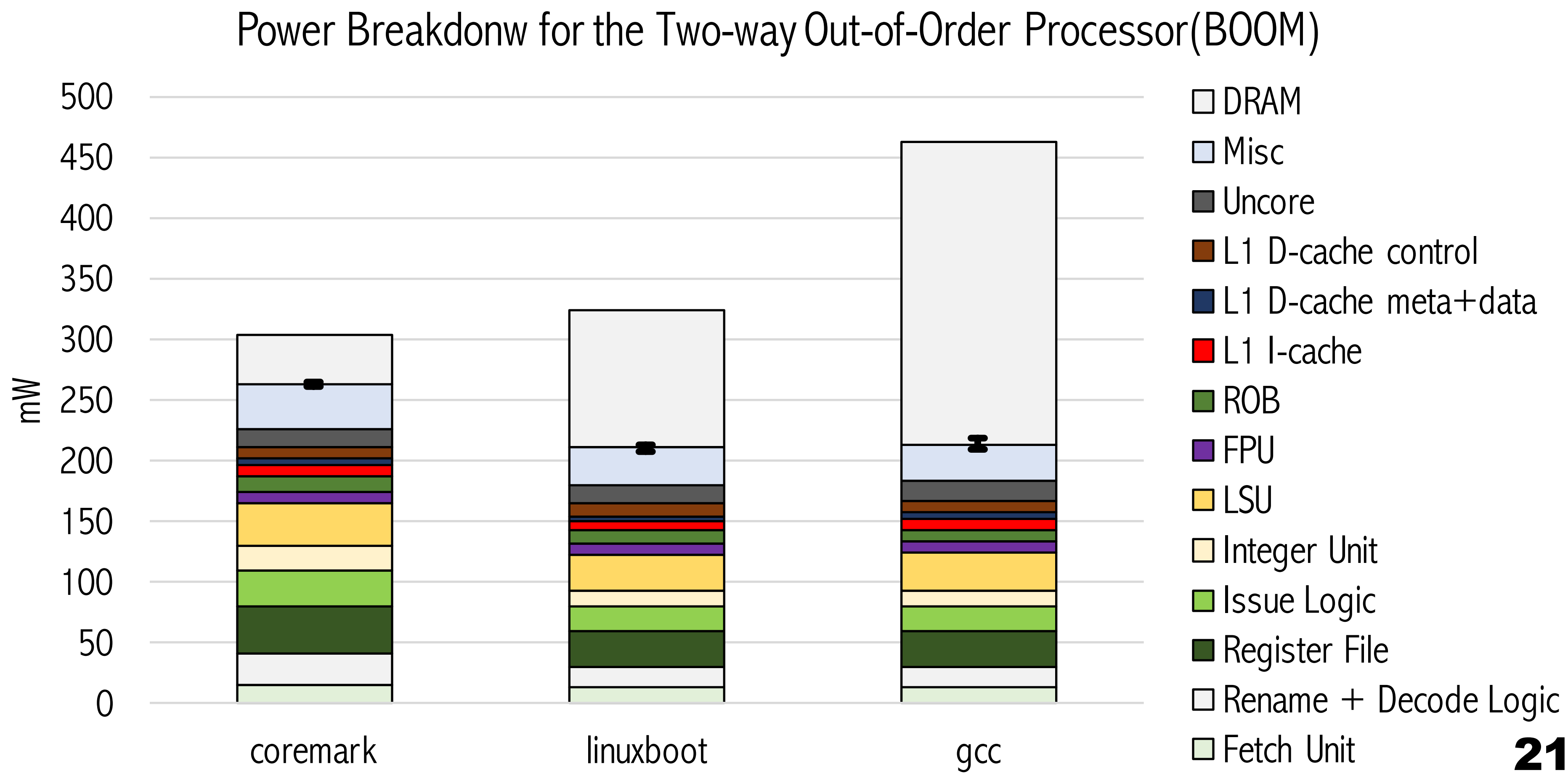
Evaluation: Power Break Down

- **Coremark**: designed to fit in L1 caches and stress processor's integer pipeline
- **Linuxboot, gcc**: larger memory footprint to stress memory system
- Replay 30 random sample snapshots of 1024 cycles



Evaluation: DRAM Power

- Micron's LPDDR2 SDRAM S4
  - 8 banks, 16K (16 x 1024) rows for each bank
- Event counters read out from scan chains
- Spreadsheet power calculator by Micron



Conclusion

Open-source this summer: Strober.org

- ***Any novel hardware design*** written in Chisel
  e.g. Accelerators for deep learning
- ***No modeling*** is necessary
  ➔ Accurate Energy Evaluation
- ***Orders of magnitude speedup*** over existing methodologies
- ***Automatically*** generates FPGA Simulators for ***performance evaluation and RTL state snapshotting***
- Operates with industry-standard CAD tools[*] to ***replay RTL snapshots for average power***

* ***Widely available to academics through academic licensing programs***

Acknowledgements

- Funding:
  - DARPA Award Number HR0011-12-2-0016
  - The Center for Future Architecture Research, a member of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA
  - ASPIRE Lab industrial sponsors and affiliates: Intel, Google, HPE, Huawei, LGE, Nokia, NVIDIA, Oracle, and Samsung
  - Kwanjeong Scholarship