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

<table>
<thead>
<tr>
<th></th>
<th>Analytic Power Modeling + µarch Software Simulation</th>
<th>RTL / Gate-level Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g.</td>
<td>McPAT, Wattch + GEM5</td>
<td>Synopsys VCS</td>
</tr>
<tr>
<td>Accuracy</td>
<td>? [1] (Validation against real systems)</td>
<td>✓</td>
</tr>
<tr>
<td>Generality</td>
<td>✗ (Additional development efforts)</td>
<td>✓</td>
</tr>
<tr>
<td>Speed</td>
<td>✗ (&lt; 300 KHz)</td>
<td>✗ (&lt; 1 KHz)</td>
</tr>
</tbody>
</table>

[1] Xi et. al. “Quantifying sources of error in McPAT and potential impacts on architectural studies”, HPCA `15
The Strober Framework

Custom Transforms

Chisel RTL

CAD Tools

FPGA Simulation

Gate-level Simulation

Post-layout Designs

Full Program Execution

I/O Traces

RTL State Snapshots
Chisel: Constructing Hardware In a Scala Embedded Language

chisel.eecs.berkeley.edu

- 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
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*

---

**Diagram:**
- **Matching Points**
- **Verification Tool**
- **Chisel RTL**
- **Chisel Verilog Backend**
- **Verilog RTL**
- **Logic Synthesis / Place & Route**
- **Post-layout Design**
- **Gate-level Simulation**
- **Samples**
- **Custom Transforms**
- **FPGA Simulation**
**Problem:** Slow RTL state loading on gate-level designs with simulation scripts (e.g. Synopsys UCLI scripts) – 400 commands / sec $\rightarrow$ 80 sec for 35k flip-flops

**Solution:** Custom **VPI routines** to load RTL state – 20,000 commands / sec $\rightarrow$ 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
## Target Design: Rocket Chip Generator

[GitHub Link](https://github.com/ucb-bar/rocket-chip.git)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fetch-width</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Issue-width</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Issue slots</td>
<td></td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>ROB size</td>
<td></td>
<td>24</td>
<td>32</td>
</tr>
<tr>
<td>Ld/St entries</td>
<td></td>
<td>8/8</td>
<td>8/8</td>
</tr>
<tr>
<td>Physical registers</td>
<td>32(int)/32(fp)</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td>L1 I$ / D$</td>
<td>16 KiB / 16 KiB</td>
<td>16 KiB / 16 KiB</td>
<td>16 KiB / 16 KiB</td>
</tr>
<tr>
<td>DRAM latency</td>
<td>100 cycles</td>
<td>100 cycles</td>
<td>100 cycles</td>
</tr>
</tbody>
</table>


How Fast Is Strober?

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

<table>
<thead>
<tr>
<th>Execution Length</th>
<th>100B cycles</th>
<th>FPGA Synthesis Time</th>
<th>1 hour</th>
</tr>
</thead>
<tbody>
<tr>
<td># of snapshots</td>
<td>100</td>
<td>FPGA Clock Rate</td>
<td>50 MHz</td>
</tr>
<tr>
<td>Replay Length</td>
<td>1000 cycles</td>
<td>FPGA Simulation Speed</td>
<td>3.6 MHz</td>
</tr>
<tr>
<td># of Instances of Gate-level Simulation</td>
<td>10</td>
<td>Gate-level Simulation Speed</td>
<td>12Hz</td>
</tr>
</tbody>
</table>

- 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 X (1000 cycles / 12 Hz + 2.5 min) / 10)
  - Strober Total = **9.4 hours**
Underlying Theory: Central Limit Theorem

- No. Samples Independent of Execution Length
- Longer Execution, More Speedup
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

Power Breakdown for the Two-way Out-of-Order Processor (BOOM)
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

Power Breakdown for the Two-way Out-of-Order Processor (BOOM)
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