EE290C Project Proposal: Performance and Energy Characterization of Gemmini, BOOM, and NVDLA Using State-Sampling from Firesim FPGA Simulation

Vighnesh Iyer, Billy Chau

1 Background and Motivation

The architecture and VLSI literature has produced a litany of dedicated ML accelerators over the last 3 years. These accelerators exploited unique dataflows, weight/activation sparsity, integer arithmetic, novel circuit techniques, or other techniques to achieve a perf/watt advantage over a baseline.

However, architecture papers typically measure the energy consumption of the proposed accelerator against a baseline using a (Python/C++) model of their accelerator, and thus produce unreliable energy numbers. Some papers use HLS to generate RTL implementations of different architectures. The RTL is synthesized and RTL simulation is used to produce activity traces for energy estimation. Unfortunately, RTL simulation is slow, and thus only small DNNs can be evaluated.

On the other hand, chip/VLSI papers measure the energy consumption of a taped out chip, but do not have a baseline accelerator implementation taped-out to compare against. Furthermore, once a chip is taped out, the power consumption cannot be measured for individual parts of the accelerator (at module-granularity) and cycle-by-cycle energy numbers cannot be extracted.

2 Prior Work

STROBER[6] is a fast and cycle-accurate sample-based energy estimation framework that automatically transforms RTL into a deterministic FPGA simulator. The work instruments the RTL design with a shadow scan chain that enables μArch state extraction by the host. To estimate energy, STROBER exploits the central limit theorem by taking state snapshots of the RTL at intervals using reservoir sampling and replaying those snapshots on a gate-level power simulator.

Simmani[5] is a framework that trains a power model that takes toggle activity for a small subset of signals in the RTL design (found via clustering on signal toggle densities from training VCDs), instruments the RTL design with activity counters for these signals, and enables runtime power estimation. This work demonstrated good power accuracy on running SqueezeNet on Rocket with a Hwacha vector accelerator.

DESSERT[4] is a framework for FPGA-accelerated RTL debugging that enables synthesized assertions and full μArch state dumping. This work demonstrates the flexibility of a instrumented scan chain to dump arbitrary state that we could use to “instrument” the RTL design in software after a bitstream has already been created.

3 Proposal

We propose a framework that enables:

- RTL-fidelity simulation of ML accelerators over full DNN inference passes

  We plan to re-use the Firesim FIRRTL passes that automatically transform arbitrary RTL to a deterministic, cycle-accurate model that executes on an FPGA.

- Gate-level fidelity energy modeling (with module-level granularity)

  We plan to modify and add passes to Golden Gate that add a stitched scan chain and SRAM hijack ports that allows the host to inject and pull all μArch state to and from the DUT. This technique is distinct from DESSERT and STROBER since those works use a shadow scan chain that cannot inject state into the DUT, and may be more resource intensive.

  This allows us to advance simulation time, pause at random intervals, and dump the DUT state to the host, similar to STROBER. The randomly sampled RTL-level activity traces can be formally mapped into gate-level traces and fed into Voltus with a gate-level synthesized netlist for cycle-by-cycle energy estimates.

- Performance instrumentation and bottleneck analysis (PE utilization, memory traffic analysis)

  We plan to use the auto-instrumentation (AutoCounter) feature in FirePerf[3] to dump design-specific performance counters to evaluate bottlenecks in DNN inference. We also plan to instrument the off-chip memory interface to record the DRAM address and transfer size patterns.

- Energy estimation of DRAM accesses

  The memory access trace can be replayed in DRAMSim2[7] to extract average power numbers across time. DRAM access energy is often neglected in RTL-level power analyses and isn't done in the STROBER flow.

- Exploration of the impact different dataflows and tiling patterns on energy and performance

  Finally, once the framework described has been implemented, it can be used to guide software optimizations.

We plan to use this framework to evaluate Gemmini[2] and BOOM, and if time permits, NVDLA on Firesim[1].

4 Project Infrastructure

For the initial phase of the project, we don't require any special infrastructure apart from our laptops and the BWRC servers for Genus/Jasper/Voltus (we hope VPN works). Once software simulation is successful, we will use some AWS F1 instances.

5 Project Timeline

1. **Checkpoint 1 (4/10)**: RTL simulation of a simple circuit (Risc/GCD) after default Firesim transformation, ASAP7 Genus synthesis of the circuit and SRAM macros, formal mapping of RTL VCD to gate-level VCD, scan-chain stitching and SRAM hijack FIRRTL pass working in RTL simulation

2. **Checkpoint 2 (4/24)**: Sample circuit simulating on F1 FPGA, μArch state dumping, initial energy estimation evaluation

3. **Final Report (5/8)**: Rocket with Gemmini simulating on F1 FPGA, DRAM interface and Gemmini-specific performance instrumentation, Gemmini energy estimation for pre-written ResNet50 and Mobilenet implementations, DRAMSim2 energy estimation

The timeline is aggressive and there may be points where we find ourselves stuck. It is likely that the custom FIRRTL passes introduce difficult to find bugs and may delay this timeline. It is not clear how to perform formal mapping of RTL VCDs to gate VCDs, but there's probably a Cadence RAK for it.

References

[1] Farzad Farshchi, Qijing Huang, and Heechul Yun. *Integrating NVIDIA Deep Learning Accelerator (NVDLA) with RISC-V SoC on FireSim.* 2019. arXiv: `1903.06495 [cs.DC]`.

[2] Hasan Genc et al. *Gemmini: An Agile Systolic Array Generator Enabling Systematic Evaluations of Deep-Learning Architectures.* 2019. arXiv: `1911.09925 [cs.DC]`.

[3] Sagar Karandikar et al. "FirePerf: FPGA-Accelerated Full-System Hardware/Software Performance Profiling and Co-Design". In: *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems.* ASPLOS '20. Lausanne, Switzerland: Association for Computing Machinery, 2020, pp. 715–731. ISBN: 9781450371025.

[4] Donggyu Kim et al. "DESSERT: Debugging RTL Effectively with State Snapshotting for Error Replays across Trillions of Cycles". In: *2018 28th International Conference on Field Programmable Logic and Applications (FPL).* IEEE, Aug. 2018.

[5] Donggyu Kim et al. "Simmani: Runtime Power Modeling for Arbitrary RTL with Automatic Signal Selection". In: *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2019, Columbus, OH, USA, October 12-16, 2019.* ACM, 2019, pp. 1050–1062.

[6] Donggyu Kim et al. "Strober: Fast and Accurate Sample-Based Energy Simulation for Arbitrary RTL". In: *43rd ACM/IEEE Annual International Symposium on Computer Architecture, ISCA 2016, Seoul, South Korea, June 18-22, 2016.* IEEE Computer Society, 2016, pp. 128–139.

[7] P. Rosenfeld, E. Cooper-Balis, and B. Jacob. "DRAMSim2: A Cycle Accurate Memory System Simulator". In: *IEEE Computer Architecture Letters* 10.1 (Jan. 2011), pp. 16–19. ISSN: 2473-2575.