Energy Estimation for ML Accelerators (from Arch)

- Novel architecture proposals -> architecture models -> model-embedded power estimation based on FU usage and SRAM/RF accesses
- Architectural models are fast, but have some downsides:
  - Architectural power models can suffer from poor accuracy caused by left out uArch details (only known once RTL has to be written) and from poor visibility (energy breakdown by module and cycle) (e.g. Eyeriss v1 energy model)
- ML accelerator perf/power models are typically OK in terms of accuracy since DNN execution can be statically scheduled and kernels are regular
  - e.g. Eyeriss v2, SCNN
- However, integration with an SoC (OS + SMT + cache hierarchy) and data-dependent energy models complicate things and makes the arch power model unsuitable for system-level energy estimation

Energy Estimation for ML Accelerators (from Chip)

- Sometimes an architectural innovation is taken all the way to a tapeout
- A chip is the best energy “estimate” you can get. The accuracy is ideal and the chip runs in realtime (~500 MHz)
- The downsides include:
  - Requires a full tapeout flow, fabrication, and testing (1+ year)
  - Visibility of power consumption by module/SRAM/FU is lost as usually the entire ML accelerator is in one power domain
  - Cycle-level power analysis usually isn’t possible
  - Can’t make RTL changes to see how they impact energy efficiency
- Tapeouts are unsuitable for DSE and are tricky to get right

Energy Estimation for ML Accelerators (from RTL)

- RTL implementations of architectural innovations are an intermediate step between architectural model and tapeout
- There are several ways of estimating power from RTL
- Power Macromodels + Switching Activity estimation (e.g. Wattach)
  - Several macromodels variants are used (traditional ML/regression, DNN models)
  - Switching activity correlated to heuristics (signal transition density, correlated signals)
- Gate-level vectorless analysis
  - Switching activity is estimated based on top-level IO activity factors propagated to internal gate nodes; typically not accurate and
- Gate-level vectored analysis (w/ or w/out PEX)
  - Best pre-tapeout accuracy, requires going through at least synthesis if not full VLSI flow, requires VCDs from simulation, suffers from poor performance (~1-10 Hz)

Energy Estimation Techniques

	Visibility/Granularity	Fidelity/Accuracy	Speed/Startup Time
Architectural Model	FU/Memory level, often not cycle-by-cycle	Not reliable at system-level, no golden reference	~ 10+ MHz / Instant
RTL Power Macromodel	Module or block level, sometimes cycle-by-cycle	May be within 10% accuracy for small-length stimulus	~ 1 MHz / Model Training + RTL sim
Gate-level Vectorless Power Est	Module-level, average power	Generally not accurate, activity factors often overestimated	~ kHz / Post-syn
Gate-level Vectored Power Est (w/ PEX)	Module-level, cycle-by-cycle	Signoff accuracy	~ 1-100 Hz / Post-PnR + GL sim
Tapeout/Chip	Full power domain, uS time granularity	As accurate as you can get	~ 500 MHz / Design + fab + test

FPGA-Accelerated Energy Estimation

- For evaluating Gemmini in an SoC context running full DNN inference, we want the accuracy of GL energy estimation with the speed of an arch model
- Strober (Kim et al., ISCA 2016) implemented a sample-based energy estimation technique where RTL state snapshots are captured from FPGA simulation and replayed on GL sim
  - Utilizes reservoir sampling to pick a fixed number of sample points during arbitrary length program execution, confidence intervals can be constructed on the power estimates

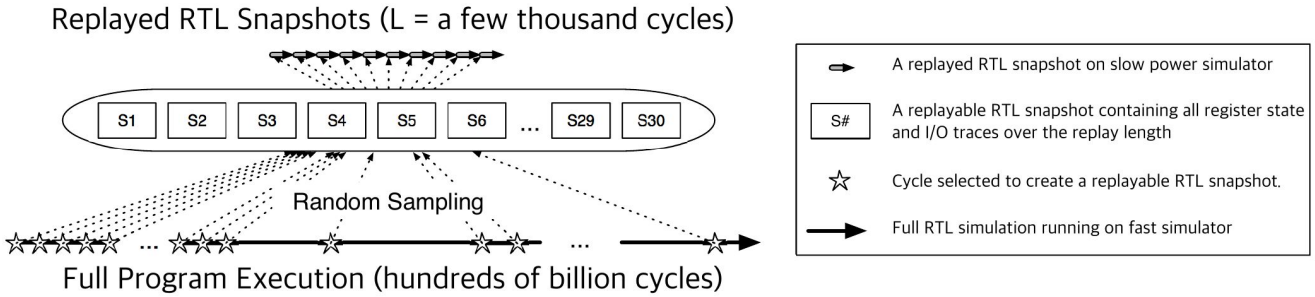
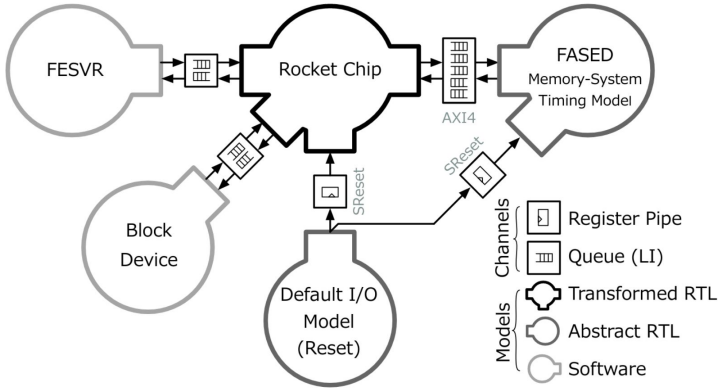


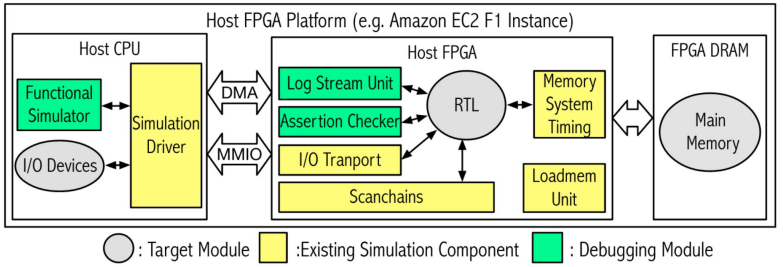
Fig. 2: Sample-Based Energy Simulation Methodology

FPGA-Accelerated Energy Estimation

- Firesim (Karandikar et al., ISCA ‘18) provides infrastructure for cycle-accurate, deterministic RTL execution on FPGAs (with timed host and DRAM interaction)
  - Enables single-step execution of arbitrary RTL on FPGAs
- Strober was written for an earlier codebase; its functionality needs to be ported to use the current Firesim/Golden Gate APIs
  - Instrument RTL design with scan chain to enable full uArch state dumping to host
  - Replay sampled uArch state snapshots of small time intervals on gate-level energy simulation
  - Dump DRAM access traces for DRAM power estimation using DRAMSim2

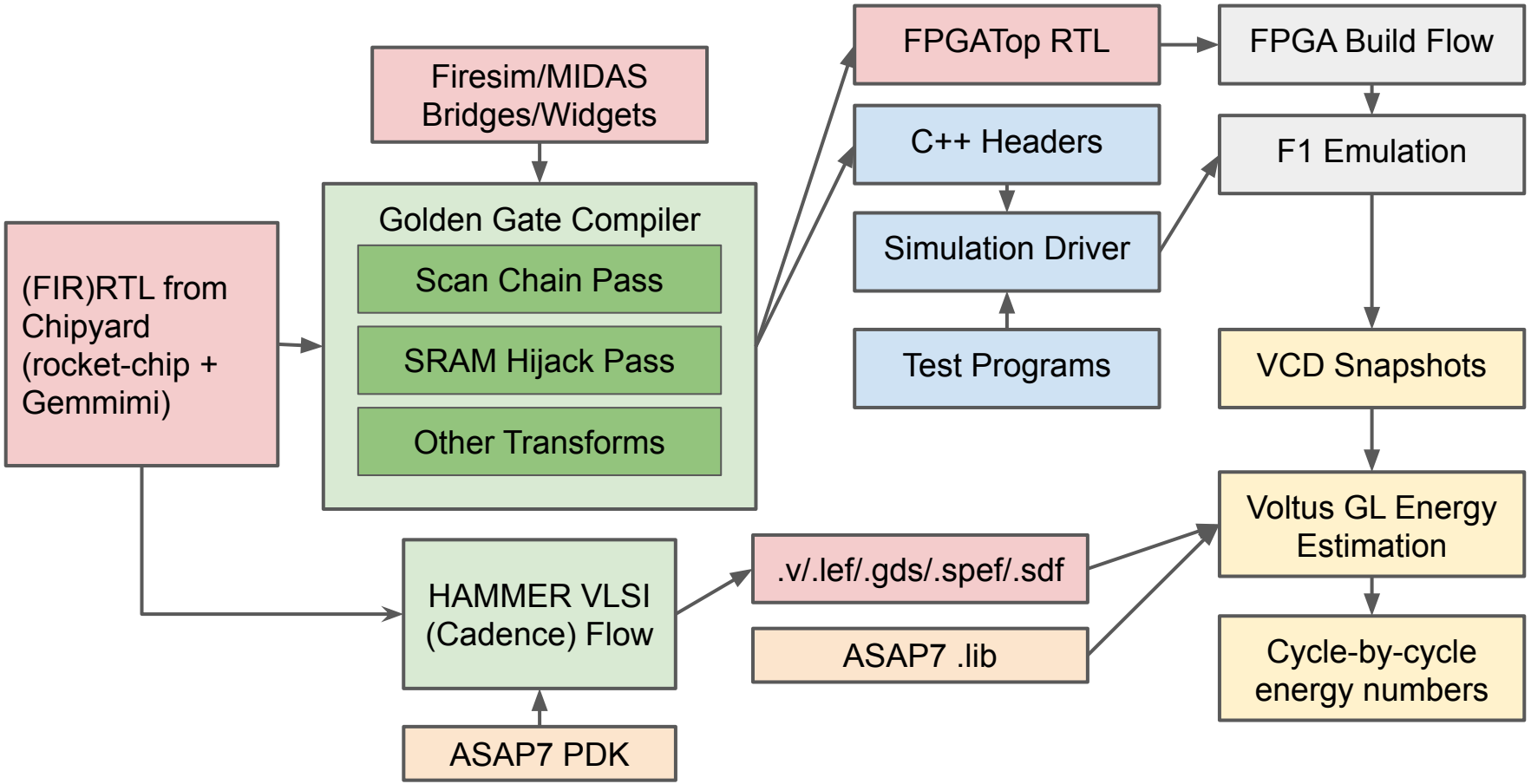


FASED, Biancolin et al., FPGA ‘19



DESSERT, Kim et al., FPL ‘18

Energy Estimation Flow



Timeline

- Strober implemented FPGA-accelerated sample-based energy estimation
  - Codebase was developed out-of-tree from Firesim and needs to be reintegrated due to internal API changes
- Aiming to demonstrate rocket-chip + Gemmini energy estimation on hand-written DNN programs (MobileNetv1, Resnet-50) by the end of May
  - This system can be used for FPGA-accelerated RTL debugging (DESSERT, Kim et al., FPL 2018)
  - Energy estimation will be used to guide SW optimizations and HW parameter DSE
- Later we will modify the instrumentation technique of Strober to use a stitched scan chain vs a shadow scan chain to enable *state injection* in addition to state dumping
  - Can be used for deterministic, uArch-level, fault injection
  - Only requires additional host software; the FPGA platform and RTL can be reused