Virtual Methodology For Performance and Power Analysis of AI/ML SoC Using Emulation

Vikas Singhal | Debdutta Bhattacharya | Ayub Khan

Background

Spectrum of AI/ML SoCs

- Vision/Speech/Pattern Recognition
- Data Center/High-Performance Compute
- ADAS
- Edge Computing
- Deep Learning Training
- Space/Military
- Cryptocurrency
- Disease diagnosis etc

Trends and Challenges

- Mobile/Edge AI inference SoCs
  - Low Latency, Power Efficient
- Data Center/Deep Learning Training SoCs
  - Scalable for training applications, High Performance
- Space/Military/ADAS
  - Fault tolerant, Safety Critical

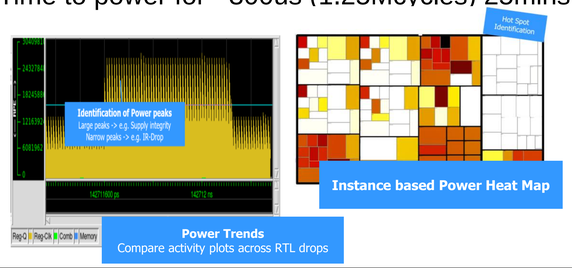
Motivation

Virtual methodology for integrated ASIC and SW stack verification for –

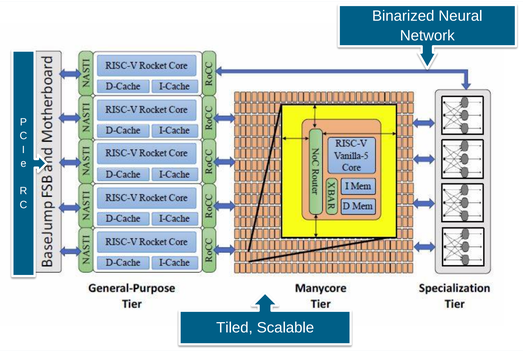
- Architectural exploration
- Power optimization
- Faster debug
- Accelerating product schedule

Power Analysis on Emulation

Power profile and hear map using emulation to identify power peaks and hot spots
Time to power for ~300us (1.25Mcycles) 25mins

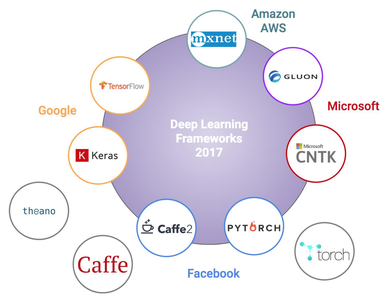


AI/ML SOC HW Verification – Challenges



Challenge	Need
Massive tiled designs	Compilation of huge SoC RTL in short time and as-is emulation model compilation
Power	Measure power while running SW applications
Efficient debug	Enable simulation-like full RTL visibility
Performance	Simulate orders of magnitude faster over software simulation

AI/ML SOC SW Validation - Challenges

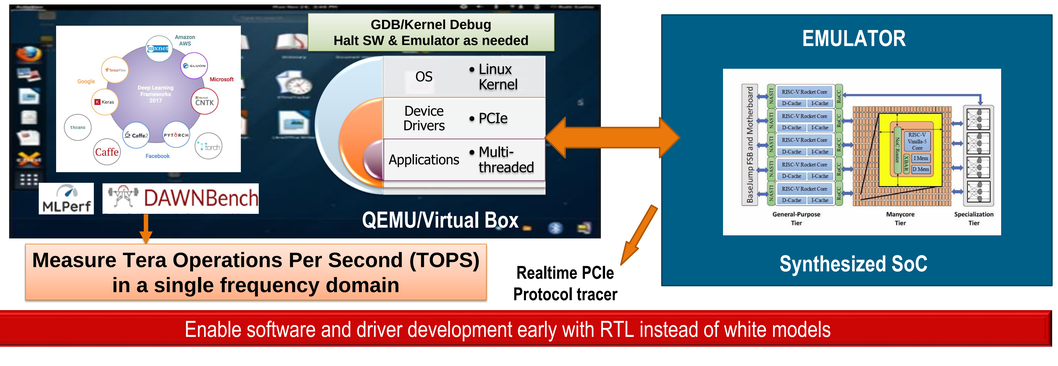


MLPerf

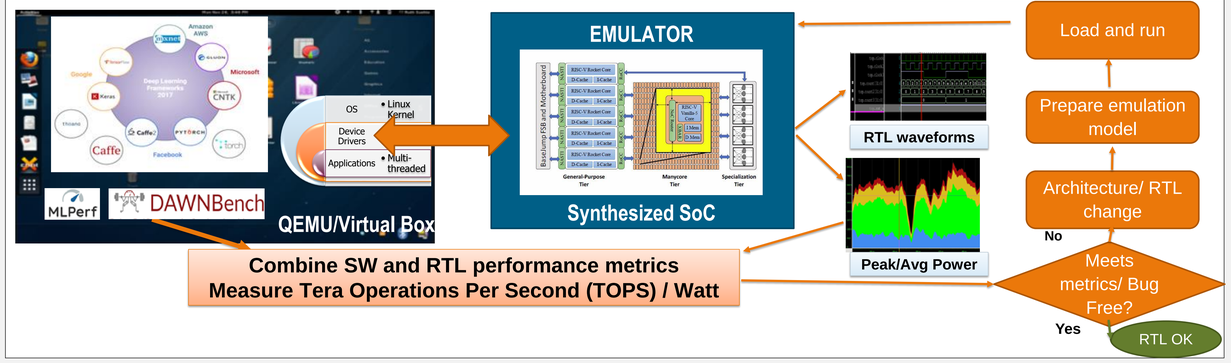
DAWNBench

Challenge	Need
Growing AI frameworks & applications	Ability to run AI frameworks built on top on variety of host OS
Performance metrics	Run AI performance benchmarks (Eg. MLPerf, DAWNBench)
Power Analysis with SW Stack	Run SW and RTL together to measure power and TOPS/Watt

AI/ML SW-HW Validation Platform



Virtual Methodology For Performance & Power Analysis of AI/ML SoC Using Emulation



Results

- **Compile Time***

Design Tile Configuration	Compile Frequency	Design Size (million gates)	Compile Time (hrs)
16x31	1785 KHz	25	0.6
64x64	1351 KHz	175	0.6
128x128	1020 KHz	698	2.1
256x256	943 KHz	2870	7.5

- **Runtime***

Test Name	Simulation Wall Clock Time	Emulation Wall Clock Time
manycore_streambuf_single_image	1141 mins	17.22 mins
manycore_streambuf_layer_7	202 mins	10.02 mins

*Open source accelerator-centric SoC with a tiered accelerator fabric targeting highly performant and energy efficient embedded systems (Reference: http://opencelerity.org)

Other Approaches

ICE (In-Circuit Emulation)	FPGA prototyping
• 2 separate domains - ICE targets running at GHz; Emulated design at MHz • Physical speed adaptors => Inaccurate performance benchmarking • HW/SW co-debug difficulty : Stopping HW clocks not possible due to asynchronous targets • Access restrictions	• Used for software development : Faster than emulation • Longer compile time & compile effort • Debug is very limited • Power estimation is a challenge

Comparison to other approaches

Metric	Proposed Method	ICE based Methodology	FPGA Prototyping
Run AI SW frameworks	**Yes**	Yes	Yes
Accurate performance benchmarks (Requires single frequency domain)	**Yes**	**No**	**No**
RTL based power analysis for long run (Requires full visibility and fast waveform upload)	**Yes**	Yes	**No**
Fast RTL turnarounds (Fast compile, run and debug times)	**Yes**	Yes	**No**
HW-SW debug : gdb + Stop HW Clocks	**Yes**	**No**	**No**

Summary and Next Steps

Proposed virtual emulation methodology for is the only solution which enables –

Performance benchmarking
RTL based power analysis and optimization
Fast RTL turnarounds
HW-SW debug using gdb and HW clock stoppage

ICE based HW emulation and FPGA prototyping offer similar capabilities but have serious deficiencies