Emulation based Power and Performance Workloads on ML NPUs

Pragati Mishra, Ritu Suresh, Issac Zacharia, Jitendra Aggarwal
Arm Ltd

arm

2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES VIRTUAL FEBRUARY 28 - MARCH 3, 2022

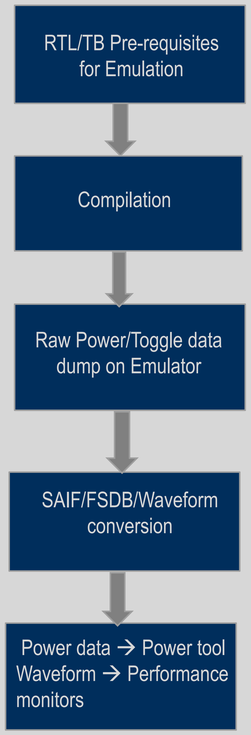
Motivation

- Simulators are used for functional/Power verification
- Complex designs on simulators are of slow process
- Power and Performance benchmarks are not efficient to run on simulators due to their nature of complexity and longer runs
- Emulation is a proven flow for modern verification, and it helps significantly to reduce turnaround in Power and Performance coverage closure

Power Analysis

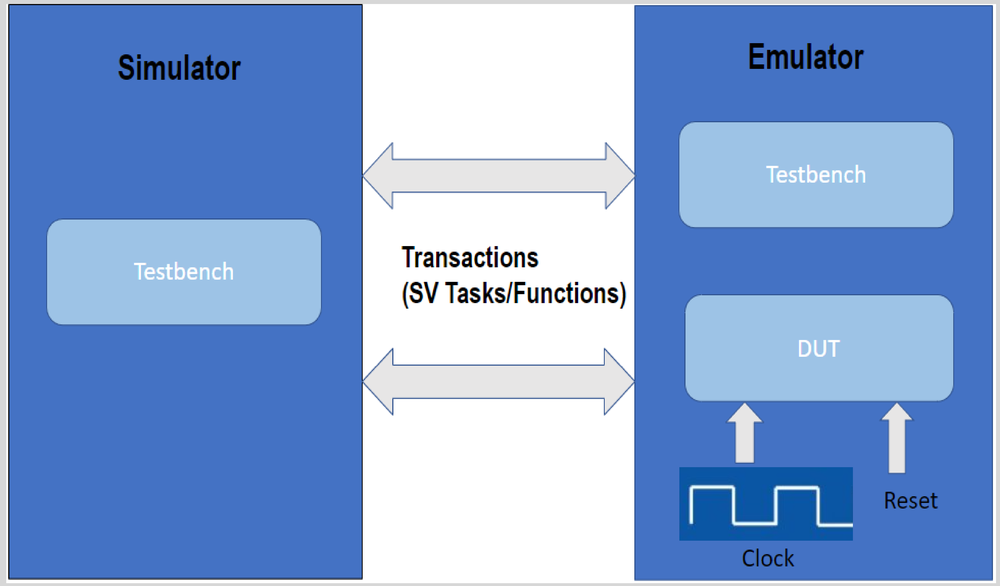
- Power validation requires real applications to run for power calculation which takes significantly longer than verification tests
- Emulation helps to Power closure with significant gain in verification timelines
- Power tools are easily pluggable with Emulation/Hardware-Acceleration platforms to replace simulators with very low error margin
- By virtue of Emulation synthesis step, one has to make sure of replacing all RTL/TB memories to synthesizable models

Methodology Flow

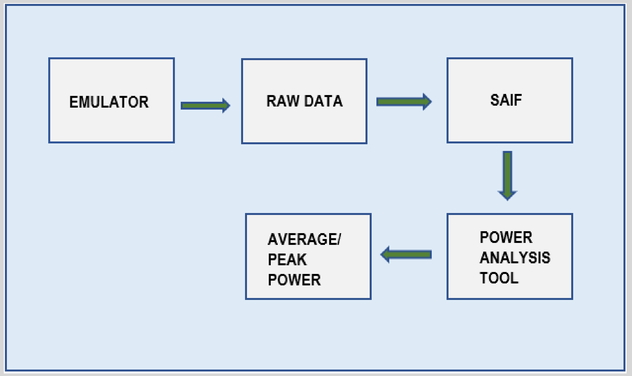


Methodology

- Accellera Co-emulation methods are adopted to migrate existing testbench to Emulators
- Transactor level modelling used to split the testbench into HDL and HVL, in which HDL contained DUT
- System verilog task and functions are used to transfer data between HDL and HVL



- After compilation of HDL and HVL, real applications are executed to dump Power (SAIF/FSDB) and Performance data (Monitors/Counters)
- Emulators dump raw toggle data from Hardware, so there is an intermediate step to convert it into SAIF/FSDB which goes directly into power tools




```
(INSTANCE u_dut
   (PORT
      (ARADDRM[0] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[10] (T0 4.90823e+07) (T1 4.91336e+07) (TC 170787))
      (ARADDRM[11] (T0 4.58078e+07) (T1 5.24082e+07) (TC 112196))
      (ARADDRM[12] (T0 4.4956e+07) (T1 5.32599e+07) (TC 63723))
      (ARADDRM[13] (T0 5.10019e+07) (T1 4.7214e+07) (TC 35441))
      (ARADDRM[14] (T0 5.38904e+07) (T1 4.43255e+07) (TC 72829))
      (ARADDRM[15] (T0 4.87415e+07) (T1 4.94744e+07) (TC 41472))
      (ARADDRM[16] (T0 5.4261e+07) (T1 4.39549e+07) (TC 28957))
      (ARADDRM[17] (T0 5.63239e+07) (T1 4.1892e+07) (TC 14824))
      (ARADDRM[18] (T0 5.02006e+07) (T1 4.80153e+07) (TC 15528))
      (ARADDRM[19] (T0 5.57005e+07) (T1 4.25154e+07) (TC 9322))
      (ARADDRM[1] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[20] (T0 5.51429e+07) (T1 4.3073e+07) (TC 5502))
      (ARADDRM[21] (T0 5.58735e+07) (T1 4.23425e+07) (TC 3826))
      (ARADDRM[22] (T0 5.06982e+07) (T1 4.75177e+07) (TC 3904))
      (ARADDRM[23] (T0 5.92732e+07) (T1 3.89427e+07) (TC 3520))
      (ARADDRM[24] (T0 6.4588e+07) (T1 3.36279e+07) (TC 1238))
      (ARADDRM[25] (T0 6.02824e+07) (T1 3.79335e+07) (TC 1408))
      (ARADDRM[26] (T0 5.32832e+07) (T1 4.49328e+07) (TC 1858))
      (ARADDRM[27] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[28] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[29] (T0 1.23028e+07) (T1 8.59131e+07) (TC 2224))
      (ARADDRM[2] (T0 9.82158e+07) (T1 90) (TC 4))
      (ARADDRM[30] (T0 1.23028e+07) (T1 8.59131e+07) (TC 2224))
      (ARADDRM[31] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[32] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[33] (T0 9.82159e+07) (T1 0) (TC 0))
      (ARADDRM[34] (T0 9.82159e+07) (T1 0) (TC 0))
```

SAIF Generated from Emulator

```
dpa -<raw data dump>
source ../dpa_fsaif.list
dpa -saif -addscope {hierarchy to dump}
dpa -addinst -depth 0 {hierarchy to dump}
set probeCount [dpa -list -max 1]
dpa -outfile -saif SAIF/dpa_capture -instance {instance name} -segment 100%
dpa -upload
date
exit
```

Offline SAIF Conversion Script

Challenges and Solution

- To understand the correct boundary of simulator testbench to split into HDL and HVL, Accellera standard based co-emulation technique is adopted
- To verify testbench change with full regression suites and debugs around HDL-HVL boundary, Emulator debug features are used
- To optimize Emulation modelling, EDA tools are tuned based on ML NPU design complexities
- Power benchmarks involved a validation of mismatch in signal toggles between simulation and emulation, switching off emulator logic optimization engine
- Performance benchmarks run for billions of cycles and occupies the emulator hardware for days, hardware stability is taken care by EDA
- To transfer raw emulator toggle data directly to Power tool, work is ongoing i.e. Online streaming mode

Results and Next steps

- First milestone was of 330x runtime gain in emulation over simulation runtime
- Additional 2x gain (total 600-800x) was attained by optimizing HDL-HVL boundary communication (memory read/write calls are clubbed and few redundant calls are removed)
- Largest Power vector on simulator takes several days which came down in few hours on emulator
- Performance benchmarking was a challenge on simulator due to long runtime for billions of cycles, which was easily overcome on emulation
- Next step is:
  - To unify testbench to avoid maintenance of separate testbenches for simulation and emulation
  - To ensure scalability of existing efforts in future projects

Tests	Simulation (in seconds)	Emulation (in seconds)	Optimized TB in Emulation (in seconds)	Overall Gain (Simulation vs Emulation)
Test 1	233352	692	385	600x
Test 2	344232	1264	493	700x
Test 3	257075	1036	458	560x
Test 4	374534	1020.08	488	760x
Test 5	381338	900	441	860x