Emulation based Power and Performance Workloads on ML NPUs

Pragati Mishra, Arm Ltd , Bengaluru, India (pragati.mishra@arm.com)

Ritu Suresh, Arm Ltd , Cambridge, UK (ritu.suresh@arm.com)

Issac P Zacharia, Arm Ltd, Cambridge, UK (issac.zacharia@arm.com)

Jitendra Aggarwal, Arm Ltd, Bengaluru, India (jitendra.aggarwal@arm.com)

***Abstract*— Modern SoCs are touching new heights in terms of size and complexity. With this, the demand for low-power devices has also increased. Hence, Power Verification has become essential, and it is important to keep the power numbers under limits. To achieve this, Power calculations need to be performed from the early stages of the design cycle.**

Software Simulators were used for this purpose, but for large and complex designs, simulators slow down. That is where emulators come into the picture. In Emulation, verification is done on hardware that can be FPGA-based or Processor-based, hence they are much faster than Simulators. Emulators significantly help to reduce turnaround in Performance and Power coverage closure.

In this paper, we will discuss the design being migrated from a simulation platform to an emulation platform and the benefits of using Emulation for Functional Verification and Power Verification.

Keywords— Emulation, Power Verification, Functional Verification

I. INTRODUCTION

Power Analysis or Power Verification has become increasingly popular these days. Due to the increase in design complexity, it is important to determine the correct stimulus to calculate average and peak power for power analysis. Average power calculation is important to determine the battery life of a device. The average power is generally calculated by capturing the toggle activity like SAIF files and then using power analysis tools for its computation. Peak power analysis is done by determining the hot spots in the design by running it for billions of cycles. Peak power is calculated, not by determining the toggle counts at each cycle, but by a sampling ratio defined by the user. The peak areas are narrowed down, and detailed power analysis is done which helps in accurate peak power calculations.

Due to shrinking technology nodes and compact designs, simulators have not been a preferred way for verification. Simulators are slower than Emulators, emulators run at a much faster speed and can run billions of cycles, thus they can cover the corner cases of failure in the design which is not possible in simulation. Running tests for larger cycles helps in calculating the toggle counts in the design that may be helpful for average power analysis.

II. POWER ANALYSIS: SIMULATION VS EMULATION

With the increasing complexity of designs, power verification has become as important as functional verification. The need for low power devices and reducing peak power issues in the devices has made the engineers use power verification and power analysis techniques.

Power analysis is done by capturing the average and peak power consumption in the design by applying stimulus. For accurate power analysis, the tests should be run for a long time to ensure that actual power peaks are captured. Earlier power analysis was done using simulation but there were certain limitations.

Simulators can only run for a few hundreds of thousands of cycles, which is not sufficient for accurate power analysis. Therefore, emulation has become a necessary power analysis technique for today's SoCs. Hardware Emulators can run billions of cycles with a speed of MHz which is crucial in identifying the Peak Power consumption. The Power

Analysis tools are easily pluggable with Emulation or Hardware Acceleration platform. Hence, Emulators can run real-world stimuli for hundreds of millions of cycles which ensures the accuracy of Power Calculations. They produce SAIF files for power computation. The SAIF file contains the toggle counts of all the signals in the design. This SAIF can be supplied to power analysis tools for calculating average power. The results obtained from emulation are closer to actual power consumption. Emulation speed allows designers to run power analysis several times before tape out which is not possible in simulation.

III. METHODOLOGY

In the case of Arm NPUs, simulators became a bottleneck for computing power and performance workloads. The tests running on simulation took several weeks to complete and sometimes even a month. Hence, it was decided to migrate the design to an emulation platform.

The existing setup was based on simulators. The top module consisted of the DUT instantiation and other testbench modules.

To migrate the design to an emulation-based environment, the testbench was split into a dual top hierarchy, which is, Hardware top and Software top. The Accelera co-emulation methodology was used during the migration in which the synthesizable part of the design runs of the emulator and the non-synthesizable part of the design runs on the simulator and the communication between the HW top and SW top is established using transactors or signals. While re-modeling the TB, The DUT and synthesizable modules of the TB were migrated to the HW top, while the rest of the modules which were non-synthesizable constituted the SW top.

Transaction-based communication approach was used to establish the connection between HW and SW i.e., using system Verilog tasks and functions. For each task or function call on the SW side, there were corresponding tasks and functions on the HW side which constituted hierarchical references for the HW-SW communication.

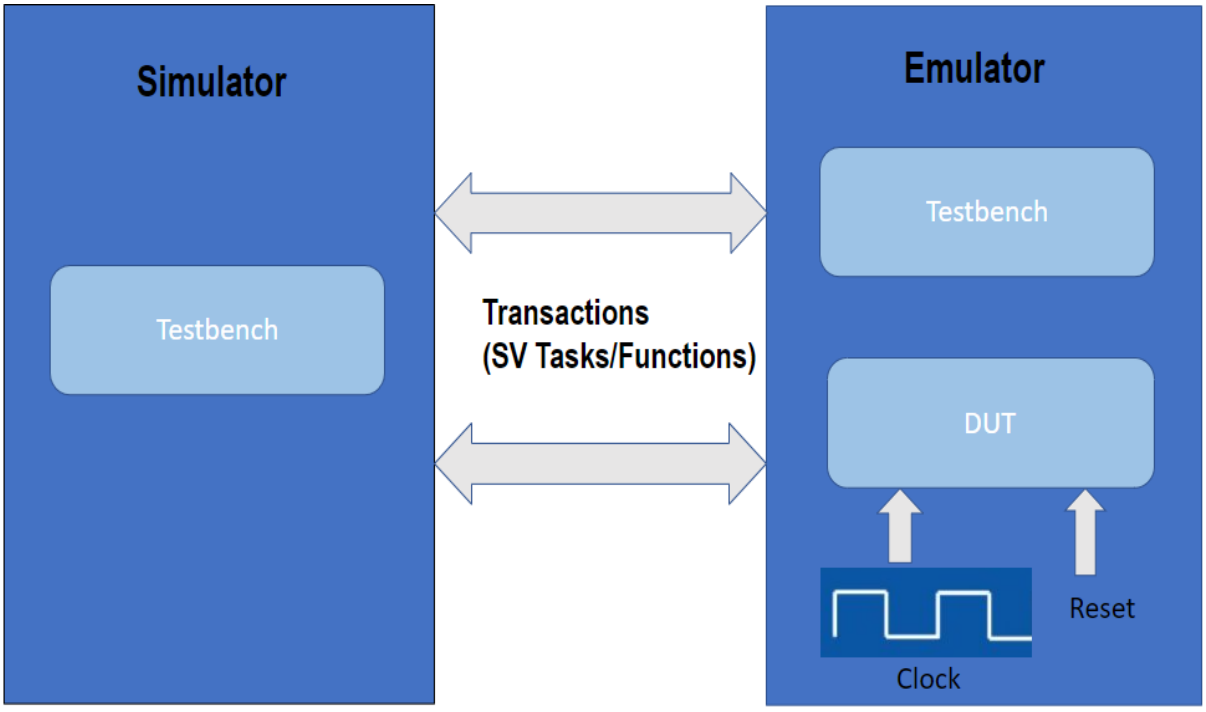
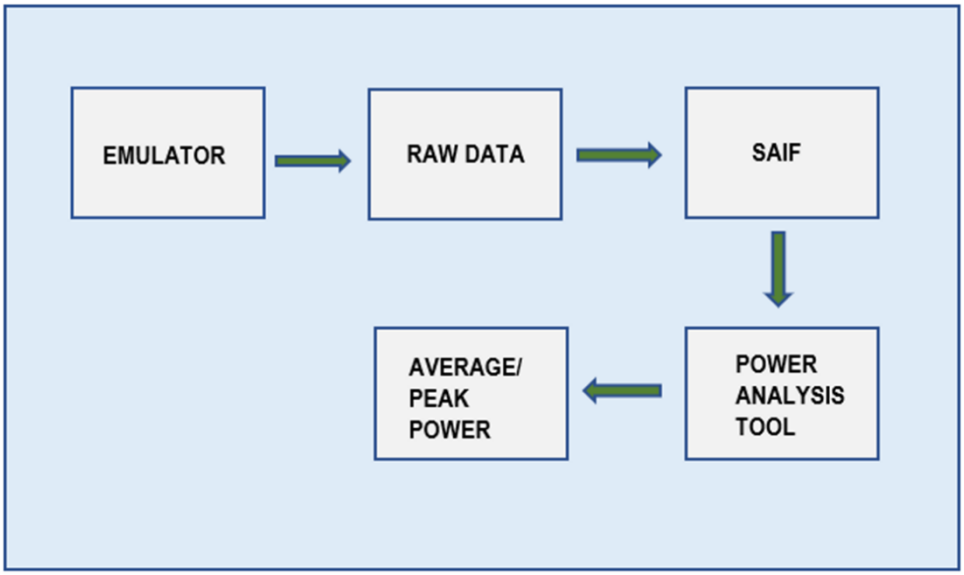


Figure 1. Co-Emulation

After the migration of the design on the emulation platform, a low throughput was observed. Hence, the testbench was further optimized to improve the emulation throughput. There were a few redundant software calls on the testbench which were reduced by increasing the transaction buffer size and a huge improvement was achieved.

After compilation of HDL and HVL, real applications are executed to dump Power (SAIF/FSDB) and Performance data (Monitors/Counters)
The Performance Analysis was enabled by adding NPU counters in RTL to do bandwidth calculation at various layers in the pipeline. We dumped counter data and used scripts to do a breakdown of per-layer data in simulations of neural network executions, enabling us to get per-layer data on bandwidth, various hardware utilization metrics.

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. Since the wave dumps or SAIF dumps for a large design could be time-consuming and would occupy the hardware emulator for a long time, the technique adopted was dumping the raw data during the emulation run and then the raw data was converted into SAIF offline, without any dependency on hardware emulator. An efficient flow was set up for this, the script used for offline SAIF conversion was specific to each emulator and consisted of forward SAIF files as an input along with the details required for the SAIF dump such as hierarchy, output dump file information, and other tool-specific options. The SAIF generated was then used as an input to the power analysis tools for power computation.



The Power measurements were done on post-route netlist since RTL-based power measurements have been less accurate. 0-delay power measurements were run on the platforms to calculate average power. The tests have been designed to enable all computational resources during the measurement window, meaning, as much toggling as possible in the NPU. With this test, the ambition is to create a scenario that is a worst-case for power, which may never happen in practice. The toggle information is dumped in the SAIF file, The power is then calculated by an EDA tool, it uses toggle statics information in combination with full knowledge of the post-route netlist to calculate average power.

IV. CHALLENGES FACED AND THEIR SOLUTION

This section highlights the challenges faced during this activity and their solutions.

- Porting the design from Simulation to Emulation
    1. The migration of the design from simulation to emulation platform while maintaining the design functionality was a challenge that required understanding the correct boundary of simulator testbench to split into HDL and HVL, Accellera standard-based co-emulation technique is adopted
    2. During porting, simulation vs emulation mismatch was observed due to compiler optimizations, these were fixed after adding appropriate compiler switches during synthesis.
    3. To verify testbench change with full regression suites and debugs around the HDL-HVL boundary, Emulator debug features are used.

- Power benchmarks involved validation of mismatch in signal toggles between simulation and emulation, switching off emulator logic optimization engine
- To optimize Emulation modeling, EDA tools are tuned based on ML NPU design complexities
- Performance benchmarks run for billions of cycles and occupy the emulator hardware for days, hardware stability is taken care of by EDA
- To transfer raw emulator toggle data directly to the Power tool, work is ongoing i.e. Online streaming mode

V. LIMITATIONS

- SDF annotated power measurements are not possible in emulation since SDF are timing constructs that are non-synthesizable and are not supported by Emulators.
- RTL based power measurements have a 15-20% deviation from real Silicon. Thus, average power is calculated using post-route netlist.
- By virtue of the Emulation synthesis step, one has to make sure of replacing all RTL/TB memories with synthesizable models.

VI. RESULTS AND CONCLUSION

The results after the design were migrated to the Emulation Platform were observed to be much faster and accurate than simulation. A Performance gain of 330x was observed at run-time over Simulation. We further worked on optimizing the hardware-software communication channel and could attain another 2-2.5x with sorting out memory read and write calls efficiently. Hence, an overall gain of 600-800x was obtained. The following table shows the time the tests ran in simulation vs emulation platform and the time the tests finally ran in emulation when the testbench was optimized by reducing the HW-SW communication channel. Hence, it was observed that emulation proved to be a better verification technique.

Table 1

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

REFERENCES

[1] Gaurav Jain, Arunendra Tomar, Umesh Pratap, "Accurate and Efficient Power estimation Flow for Complex SoCs"

[2] https://www.newelectronics.co.uk/electronics-technology/power-verification-is-just-as-important-as-functional-verification-for-complex-socs/50313/

[3] https://www.techdesignforums.com/practice/technique/emulation-system-level-power-verification/

[4] Power Estimation - Semiconductor Engineering (semiengineering.com)