2021 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE OCTOBER 26-27, 2021

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 design cycle.**
Software Simulators were used for this purpose, but for large and complex designs, simulators slow down. That is where emulators come into picture. In Emulation, verification is done on hardware which can be FPGA based or Processor based, hence they are much faster than Simulators.
In this paper, we will discuss about the design being migrated from a simulation platform to an emulation platform and 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 increase in design complexity, it is important to determine correct stimulus to calculate average and peak power for power analysis. Average power calculation is important to determine the battery life for a device. The average power is generally calculated by capturing the toggle activity like SAIF files and then use 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 and 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 of low power devices and reducing peak power issues in the devices has made the engineers to 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 an accurate power analysis, the tests should be run for 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 few hundreds of thousands of cycles, which is not sufficient for accurate power analysis. Therefore, emulation has become 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. Emulation techniques ensure accuracy of power calculations by running real world stimulus for hundreds of millions of cycles. They produce SAIF file 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 case of Arm NPUs, simulation became a bottleneck for computing power and performance regressions. The tests running on simulation took several weeks to complete and sometimes even a month. Hence, it was decided to migrate the design to emulation platform. The co-emulation methodology was used during the migration in which synthesizable part of the design runs of emulator and non-synthesizable part of the design runs on simulator and the communication between them is established using transactors or signals. In this case, Dual top design was created which consisted of hardware top and software top. The DUT and the synthesizable part of testbench constituted the hardware top which ran on emulator while the non-synthesizable part of the testbench constituted software top which ran on simulator. Transaction based communication approach was used to establish the connection between HW and SW i.e. using system Verilog tasks and functions.

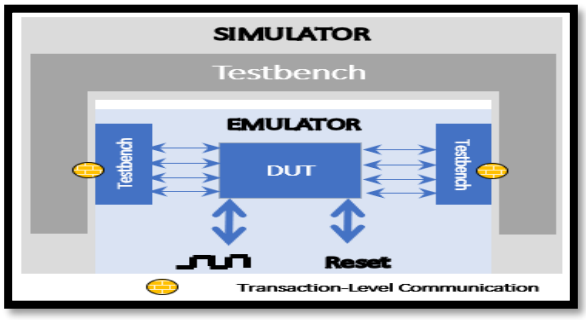

Figure 1. Co-Emulation

The testbench was further optimized to improve the emulation throughput. There were few redundant software calls on the testbench which were reduced by increasing the transaction buffer size and a huge improvement in throughput was observed. To provide the flexibility to run the design on all simulators and emulators, a unified testbench was created.

After the design was successfully migrated to emulation platform, the next step was to enable power analysis on emulation. Since the wave dumps or SAIF dumps for a large design could be a time taking process and would occupy the hardware emulator for a long time, the technique used for this was dumping the raw data on emulation while the test is running on emulator and then the raw data was converted into SAIF offline, without any dependency on hardware emulator. An efficient flow was setup 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.

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 which required splitting the design into dual top and replacing all the direct pin level access to transaction level modelling, using tasks and function calls.
    2. During porting, simulation vs emulation mismatch was observed, these issues were debugged using waves and it was found that there were few hanging nets which were optimized by the emulation

compiler, which was fixed after using certain compiler options that prevented optimization of the nets.

3. Emulator compiler was treating a for loop as a while loop and errored out during synthesis. The fix for this issue was provided in the tool by EDA vendors.

- Creating a unified testbench compatible with all simulators and emulators

  To provide the ease of using any simulator/emulator, a unified testbench was created. There were few coding style and constructs in the testbench that were not supported by some EDA tools, hence remodeling of testbench was done and then tested across all platforms.

- Longer time observed during wave and SAIF dump

  The wave and SAIF dump were taking longer than expected in emulation platform which was debugged with the EDA tool vendors and a corner case in the tool was found and fix for this was provided by the EDA vendors in the tool.

V. RESULTS AND CONCLUSION

The results after the design were migrated to Emulation Platform was observed to be much faster and accurate than simulation. A Performance gain of 320x was observed at run-time over Simulation. We further worked on optimizing the hardware software communication channel and could attain another 2x with sorting out memory read and write calls efficiently. Hence, an overall gain of 600x-700x 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 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

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)

[5] Power Analysis Needs Shift in Methodology, Power Analysis Needs Shift in Methodology - SemiWiki