Media Performance Validation in Emulation and Post Silicon Using Portable Stimulus Standard

Suresh Vasu
Multi-Media Validation Architect
Suresh.Vasu@intel.com
Nithin Venkatesh
Senior Technical Lead
Nithin.Venkatesh@intel.com
Joydeep Maitra
Senior Technical Lead
Joydeep.Maitra@intel.com
Intel Technology India Pvt Limited, No 23/56-P, Outer Ring Road, Bellandur, Bengaluru – 560103

Abstract- **The challenges of the post silicon validation are continuously increasing, driven by the higher levels of integration, increased silicon-on-chip complexity and platform performance requirements. Post silicon validation is very diverse, and the work starts well before the silicon is available in the lab. There is continuous improvement in methodologies to reduce the cost and improve debug throughput that can have huge impact in the time-to-market of the products. Based on bottlenecks of the performance validation done on multiple SOCs, we propose enhanced validation methodology using Portable Stimulus Standard. This methodology overcomes all the limitations, manual efforts done by the validation engineers and bridges the Pre-Silicon, Post Silicon and Software teams effortlessly.**

I. INTRODUCTION

The challenges of the post silicon validation are continuously increasing, driven by the higher levels of integration, increased silicon-on-chip complexity and platform performance requirements. Post silicon validation is very diverse, and the work starts well before the silicon is available in the lab. There is continuous improvement in methodologies to reduce the cost and improve debug throughput that can have huge impact in the time-to-market of the products.

Performance is the critical source of competitive advantage for modern SOCs and performance targets needs to be validated on top of the functionality. Given the multitude of ways of SOCs can be configured – different IP and interconnect topologies, numbers of masters, slaves, bus widths, packet sizes, clock speeds, the performance validation quickly gets overwhelming.

Based on bottlenecks of the performance validation done on multiple SOCs, we propose enhanced validation methodology using Portable Stimulus Standard. This methodology overcomes all the limitations, manual efforts done by the validation engineers and bridges the Pre-Silicon, Post Silicon and Software teams effortlessly.

II. VERIFICATION CHALLENGES

Figure 1 shows the percentage of total IC/ASIC project time spent in verification [1]. From the case study and analysis done by the team from Mentor, the complexity for doing the verification has increased significantly for past few years and the design size has also grown many folds. Due to many IP's are getting integrated in the SOC, verification engineers are facing lots of challenges in day-to-day debugging.

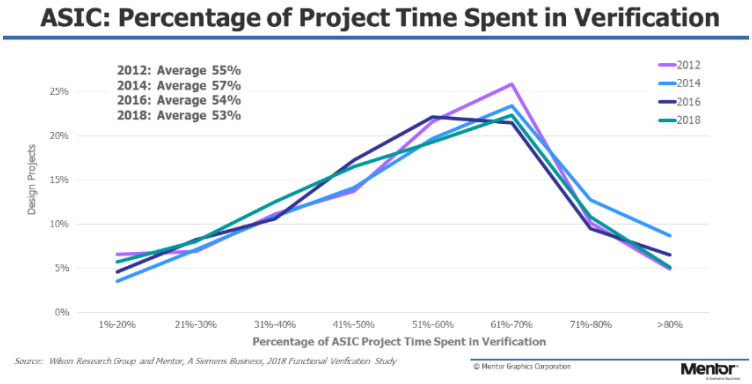


Figure 1. Percentage of IC/ASIC Project Time Spent in verification [1]

Figure 2 shows where verification engineers spend their time (on average). [1]. The study from the Mentor team shows that the verification engineer is spending more time in debugging during the chip development compared to all other verification tasks like test planning, testbench development, creating test cases, running simulation and support works. From the verification and validation engineer's experience, one of the aspects is for more debug time is due to not able to create the test content for Post Silicon Validation and no infrastructure is available to make it flexible.

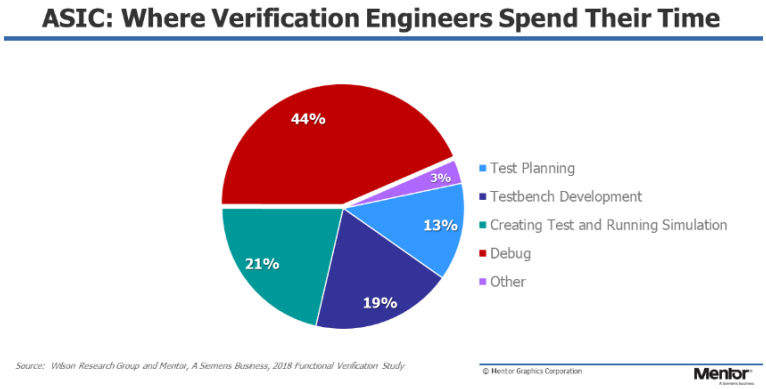


Figure 2. Where IC/ASIC Verification Engineers Spend their Time [1]

Maximum value from a PSS framework can be derived if its scalable right from IP to SOC level in terms of model re-use and test cases creation. The UVM or Formal Validation has its own limitations while we shift right in the verification and validation cycle.

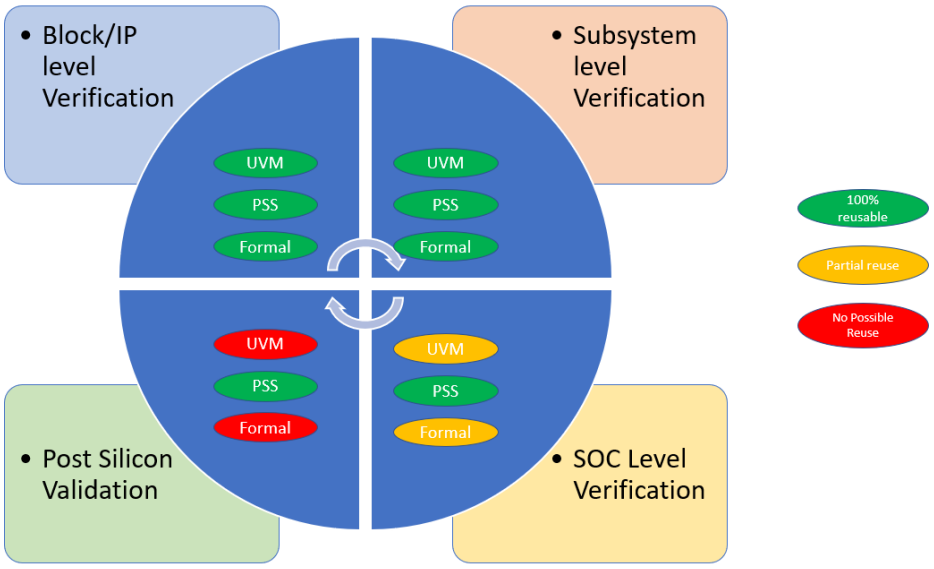


Figure 3. Verification Methodologies for an IP block

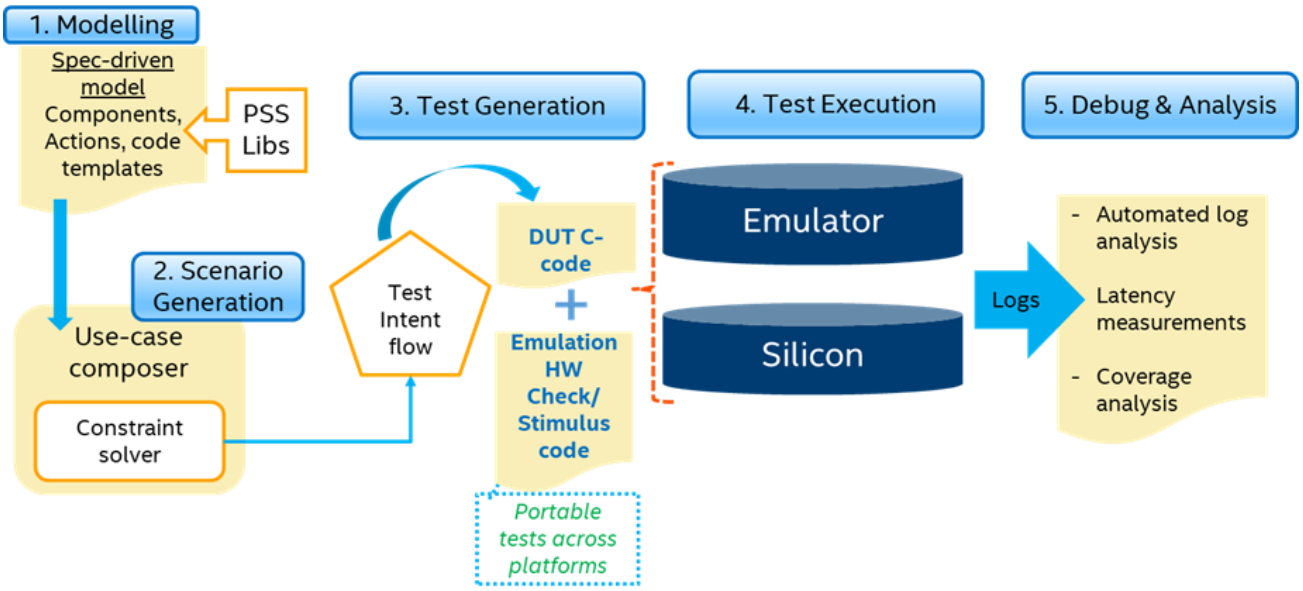
In contrast to the mature flows & methodologies ecosystem that exists for pre-silicon verification, post-silicon validation is still heavily dependent on ad-hoc test development and pointed solutions. This makes it very difficult to develop a test suite which can systematically scale and be re-used across similar/derivate projects.

Even with significant left shift in test development, preparatory execution & debug on FPGA and emulation platforms, there are often additional requirements from the software and architecture teams in late stages of a program – even after silicon arrival. A lot of such experiments require complex data paths (which mimic actual use-cases) to execute concurrently along with a controlled perturbation of system parameters. This exponentially increases the scenario space to be tested. Moreover, there are inherent limitations of each execution platform – be it FPGA or emulation (e.g. transactor model parameters) which also need to be factored in while correlating any performance measurement.

Needless to mention, driven by ever reducing time-to-market, post silicon teams are expected to provide early yet accurate power & performance feedback. This makes it imperative for such teams to have a flexible but scalable test methodology where new test development takes minimum effort. This directly impacts the time to market of the product and affects technical readiness of any proposed architectural improvements.

III. PSS

Portable Stimulus Standard defines a new test writing language that will allow automatic creation of tests targeted to different platforms from a single test source. The portable stimulus works at a higher level of abstraction that is completely independent from the type of the target platform. The test intent is represented in the form of Portable Stimulus (PS) models. These models are written in a generic way so that they can be used across multiple platforms. The Portable Stimulus models are created along with the Configuration and given to the Tool Compiler. The compiler parses these inputs and generates a visual representation of the tests in the form of graphs or flow diagram. The constrains for the tests can be applied on this visual representation and then the tests are generated for the target platform along with Graph Based Coverage.



.

Figure 4. Portable Stimulus Workflow

Figure 4 describes the PSS workflow in which the modeling, scenario generation and test generation are shown. After the constraints are resolved, the C test case are generated. The generated C test cases are validated using the Emulator/Silicon. Also, we get the debugging mechanism in the logs, latency measurement using the performance counters and the coverage data which will provide the coverage of all the features validated.

IV. Media Validation using PSS

Media subsystem consists of video/JPEG encoder and decoder blocks along with lossless compression units through which it is interfaced to the SoC NOC.

In general, for Media performance, the key metric is Macro Block/Cycles along with throughput, memory access bandwidth and frames per second for various video streams. However, for each high-level test intent, there is a huge set of variable parameters like quantization, GOP structure, Rate Disorder Optimization, Bit depth, number for frames, compression enable/disable, cache enable/disable, memory interleaving etc. Not to mention, the different formats like AVC/HEVC or even concurrent stress scenarios. Our goal is to create a test framework where the Video streams/images are the only input and the framework itself randomizes the parameters in IP C model (reference model), uses the constraint in PSS effectively and finally generates the expected test content/reference outputs.

Figure 5 show the current flow in which we have the IP C-model for Video Encoder/Decoder which will take the Raw Video (.yuv) and images (.jpg) as input and appropriates switches will be passed. All the switches are constraint and randomized in the PSS model and finally it generates the C test cases. It also generates the expected encoded or decoded golden frame data. All these bare metal C codes are compiled and validated the Media Features using the Emulation/FPGA. In the current automation, backdoor loading of the input images/video frames are also taken care. During the Post Silicon Validation, eMMC is used to load all the input images/video frames.

For the KPI performance validation, interconnect latency counters/AXI performance counters are used to measure the bandwidth and the latency numbers in the SOC. Also, using the graphical models PSS enabled to create the concurrent transcode scenarios where the encoding and decoding happens in parallel and in pipeline.

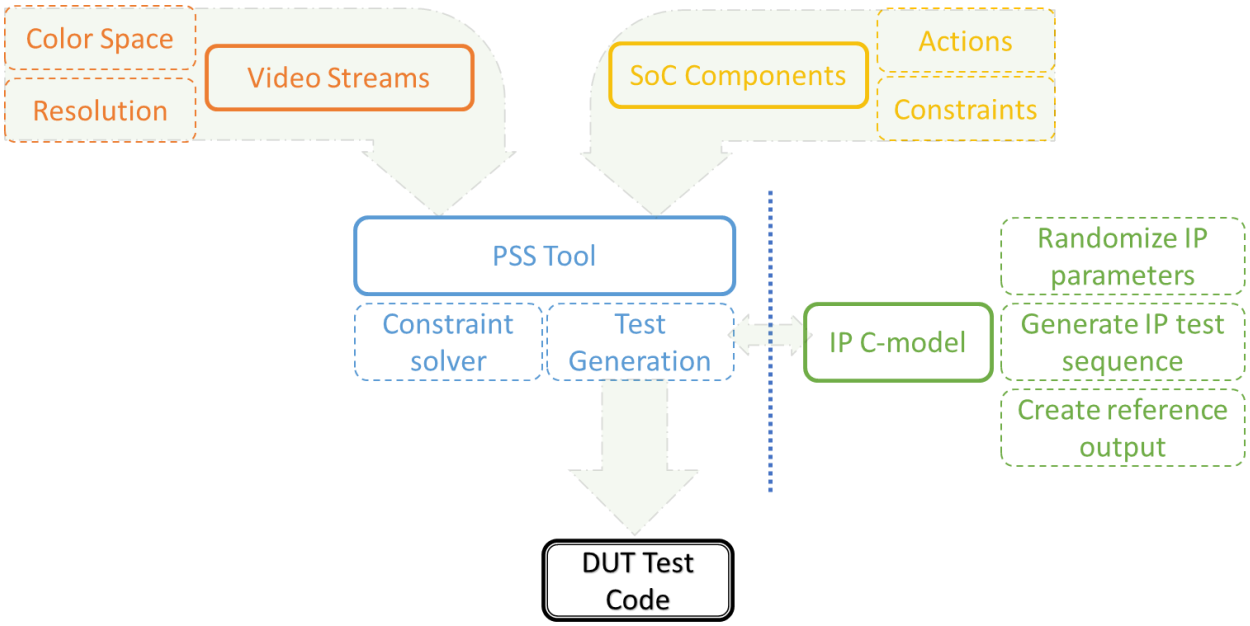


Figure 5. Media tests flow using the Portable Stimulus Standard

Figure 6 shows one of the test cases intent captured in the below graphical representation. Here we are trying to create a test case in which both Video Decoder and JPEG Decoder need to work concurrently, Video Decoder will do decoding of 1080p, JPEG Decoder will do decoding of 300x300 image size and in both the cases MMU is enabled. Rest all the parameters are randomized, and constraints are written to take care of it.

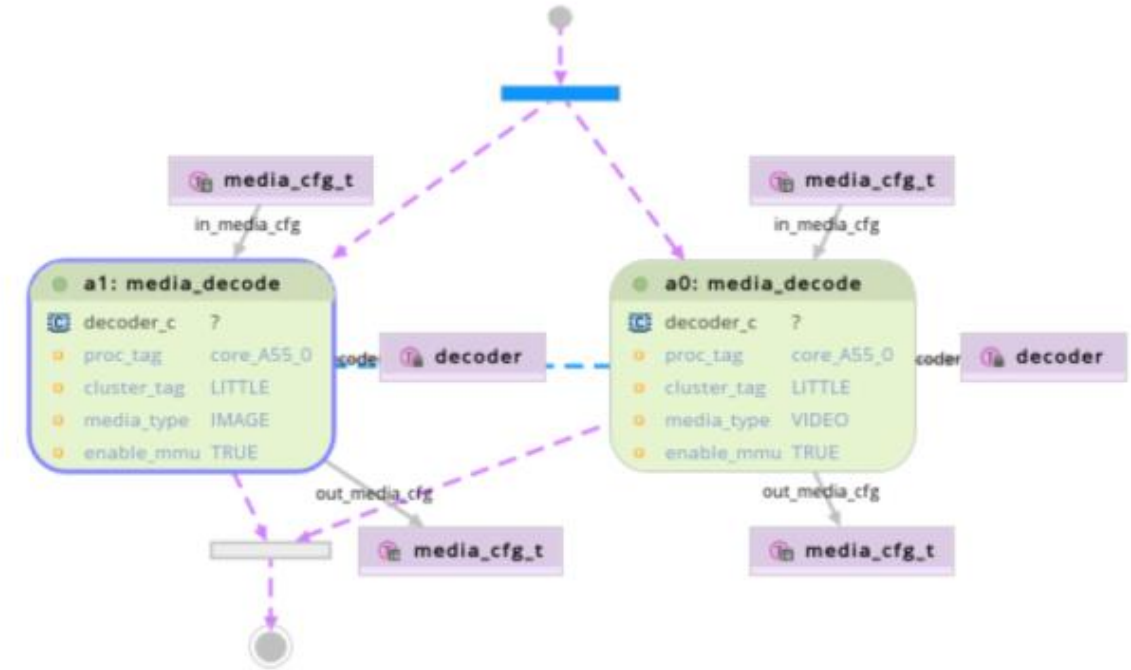


Figure 6. Media Decoder Test Intent from PSS

Figure 7 shows the generation of the test case by PSS after the constraint is resolved. During this phase, all the constraints are resolved, switches are passed to the Media C-models and C test cases gets generated. Also, the input images to preload in the memory and output expected golden images to be compared after decoding are also gets generated. In the Emulation, backdoor mechanism is used to preload the input images in the external DRAM memory.

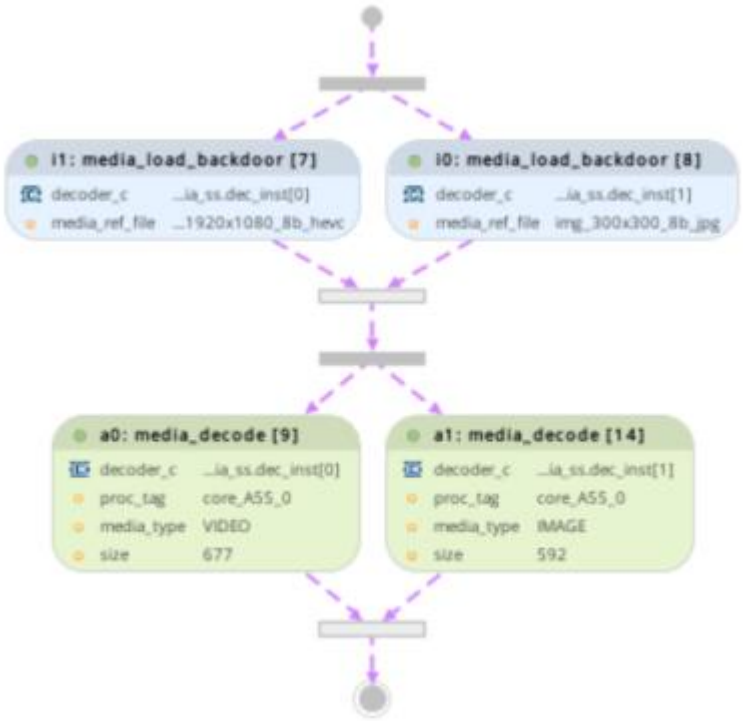


Figure 7. Media Decoder generated test case from PSS

Since all the test case intent are captured in graphical representation, it gives a lot of flexibility to the Validation engineers to generate test cases on-the-fly. For the performance validation, lots of stress/use case/concurrent/pipeline scenarios i.e. 1080p/4K/8K are generated using the PSS and able to validate the Media performance in the SOC using the Emulation. All the latency and Bandwidth in the SOC is also measured for the sign-off. The same test code can be run in the post silicon and do the performance measurement of the Media in the SOC.

For the real use cases in which we measure frames per second, the validation methodology should provide the ability to generate test cases with 30 or 60 frames data processing. Also, it should provide the ability to pipeline the scenario i.e Video Encoding followed by Video Decoding and concurrent scenarios like Video Encoding/Video Decoding in parallel. The current PSS based methodology has provided all the ability to generate complex use cases/stress scenarios to measure the Codec performance in the SOC.

A. Results

PSS based Media performance validation has cut down the content creation time by 80% as compared to the time taken in the previous projects internally. Also, the automation based on the PSS has helped to just maintain the different video streams as the inputs and generate all the test content, processed images for comparison on the fly. Also, it enabled the flexibility to enable / disable multiple IP features as per the request from software / architecture teams and able to provide the feedback very quickly. Also, it enables the coverage to sign off all the scenarios which are covered in Post-Silicon Validation.

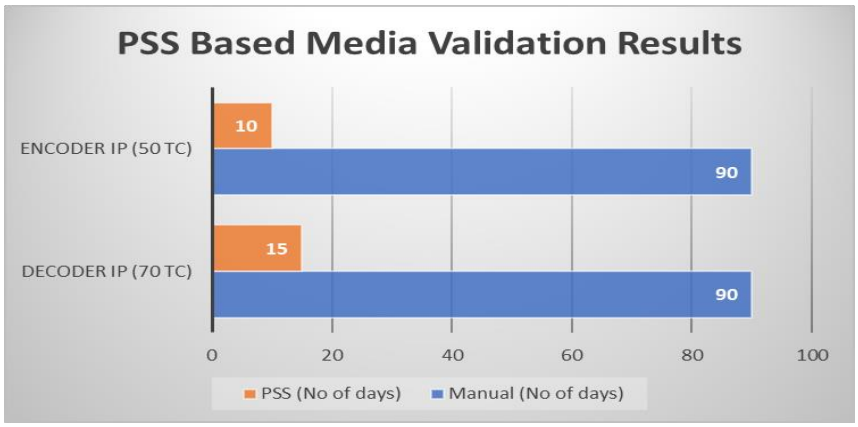


Figure 5. PSS based Media Validation Results

V. CONCLUSION

Portable Stimulus Standard is an effective methodology which will help in day-to-day validation activities, reduce the manual efforts of test content creation and save a lots of execution time. This methodology will help during the design development stage to prepare all the test content, validate in Emulation and use the same test cases in post silicon validation seamlessly. It clearly has an edge when it comes to reusability to multiple target platforms, automatic test generation to cover all graphical nodes and an additional graph-based coverage.

REFERENCES

[1] https://semiengineering.com/the-weather-report-2018-study-on-ic-asic-verification-trends/
[2] https://www.accellera.org/downloads/standards/portable-stimulus