2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE
MUNICH, GERMANY
DECEMBER 6 - 7, 2022

DVCon Europe 2022

A Novel Approach to Expedite Verification Cycle using an Adaptive and Performance Optimized Simulator Independent Verification Platform Development

Harshal Kothari, Staff Engineer, Samsung Semiconductor India Research, Bangalore, India (harshal.k1@samsung.com)

Vinay Swargam, Senior Staff Engineer, Samsung Semiconductor India Research, Bangalore, India (vinay.s9@samsung.com)

Sriram Kazhiyur Soundarrajan, Associate Director, Samsung Semiconductor India Research, Bangalore, India (sriram.k.s@samsung.com)

Somasunder Kattepura Sreenath, Director, Samsung Semiconductor India Research, Bangalore, India (soma.ks@samsung.com)

Abstract—With emergence of complex SoCs the challenges in verification has increased manifold which involves integration of pre verified IPs/Sub Systems/Partially verified subsystems using multiple vendor simulator platforms/VIPs. With time to market being a critical factor, adherence to aggressive schedules has become the new normal. Rebuilding these complex verification environments onto SoC in the given timelines is very challenging and time consuming. The Simulator Independent Verification Platform Development (SIVPD) being vendor agnostic not just addresses the issue at hand but also potentially makes use of licenses in most effective way and results in reduction of both development cycle time and saves the precious licenses cost as well.

Keywords—Simulator, verification, EDA vendor, reusable, automation

I. INTRODUCTION

SIVPD is a unique methodology which eliminates rework by enabling direct reuse of vendor/customer verification environment seamlessly. Generation of reference dumps for debug with Customer/Vendor delivered and in-house developed environment, alternative simulator environment to progress in case of a sticky issue blocking progress, divide and conquer during regression phase, effective utilization of license by optimizing runs based on availability of VIP/Simulator/Regression engine licenses, vendor agnostic run commands to ensure end user is not impacted due to multiple vendor environment are few of the planned achievements realized during the deployment (Figure 1).

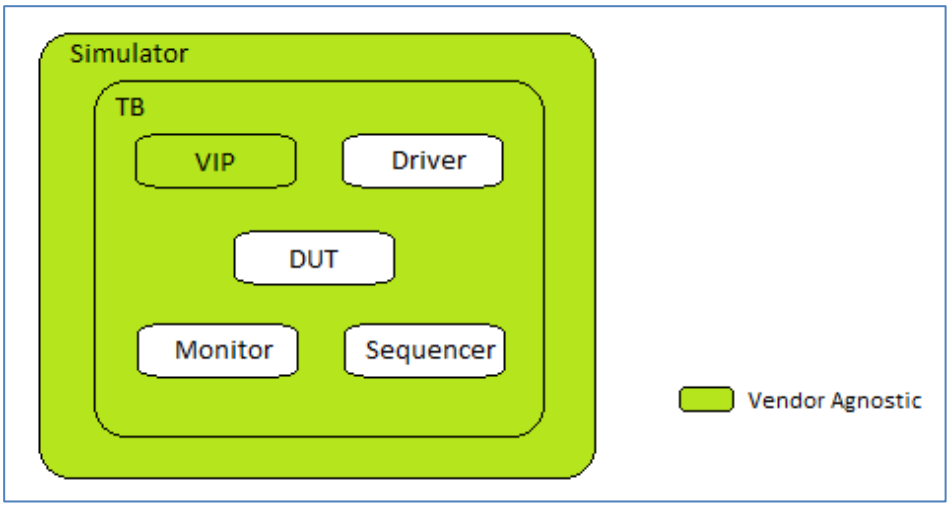

Figure 1 SIVPD Verification Environment Depicting the Vendor Agnostic Components

II. EXECUTION

Standardization was achieved by keeping a common testbench skeleton. File and directory structure inside the Verification home directory remains same, and key differentiation for simulator specific syntax and options are handled via highly flexible perl based macro configuration approach or SystemVerilog defines. As the SoC design is huge, multi-step incremental elaboration flow is adopted to ensure any change in tests or sequences does not necessitate re-elaboration of Design Under Test (DUT). The testbench generation is automated based on excel input spec. It has same look and feel on various verification scopes (IP/Sub-system/SoC) (Figure 2). This also bolsters the reusability aspect of an IP or sub-system verification environment at SoC level and vice-versa. SIVPD consists of TB related classes and 2 major modules- top and incr_top. The primary snapshot includes DUT, verilog simulation models, interface and common testbench like register definition, virtual sequencer and BUS UVC and/or VIPs which is sporadically modifiable code. Incremental snapshot consists of TB related scenarios (tests/sequence classes), interface instances and module UVC (Scoreboard/monitors/coverage etc.) During active verification cycle, top is more or less matured early whereas incr_top keeps getting frequently modified as per different use-case scenarios and verification attributes.

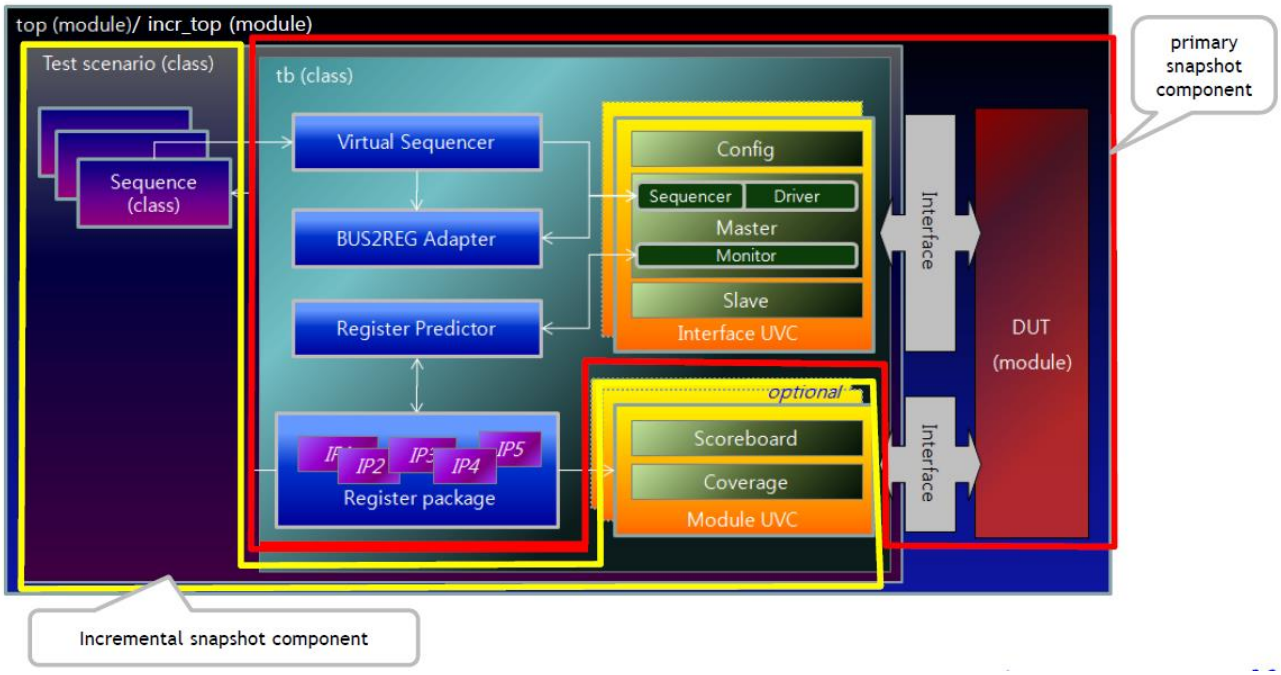

Figure 2 SIVPD Testbench

SIVPD was established as vendor agnostic by conceptualizing macro configuration for environment variables for design and verification. The run command at compile/elaboration/simulation step is parsed and macros are resolved by our proprietary in-house utility to create final run scripts which can be understood by the EDA vendor tools. These run command strings are differentiated only with tool specific switch like –<tool1> or –<tool2>, rest of the command remains unchanged for any IP/block across tools or project. Multiple design environment (DE) are created as per the IP/block use case like CPU, PCIe, USB, Display, Camera, Ethernet, Memory Interface, Flash, Low Speed IO etc. which encapsulate all verification environment (VE) or simulator related options and switches. With these DEs, the Platform Type (simulation/emulation/fpga) and Simulator Type (Tool1/Tool2) is specified in the run command. Default DE/VE/PT/ST can also be specified for projects which do not need the option of running with different simulators or environment. For the purpose of this paper, we focussed on simulations alone with Tool1 and Tool2 with RTL, Power aware RTL and Gate level netlist. After appropriate environment variables are set to specify the DUT version, build and sim directory path etc., DUT libs are compiled, top tb is elaborated, simulation snapshot is created using Save and Restart methodology and finally test scenario is run (Figure 3). Dump options are also passed along with simulation command.

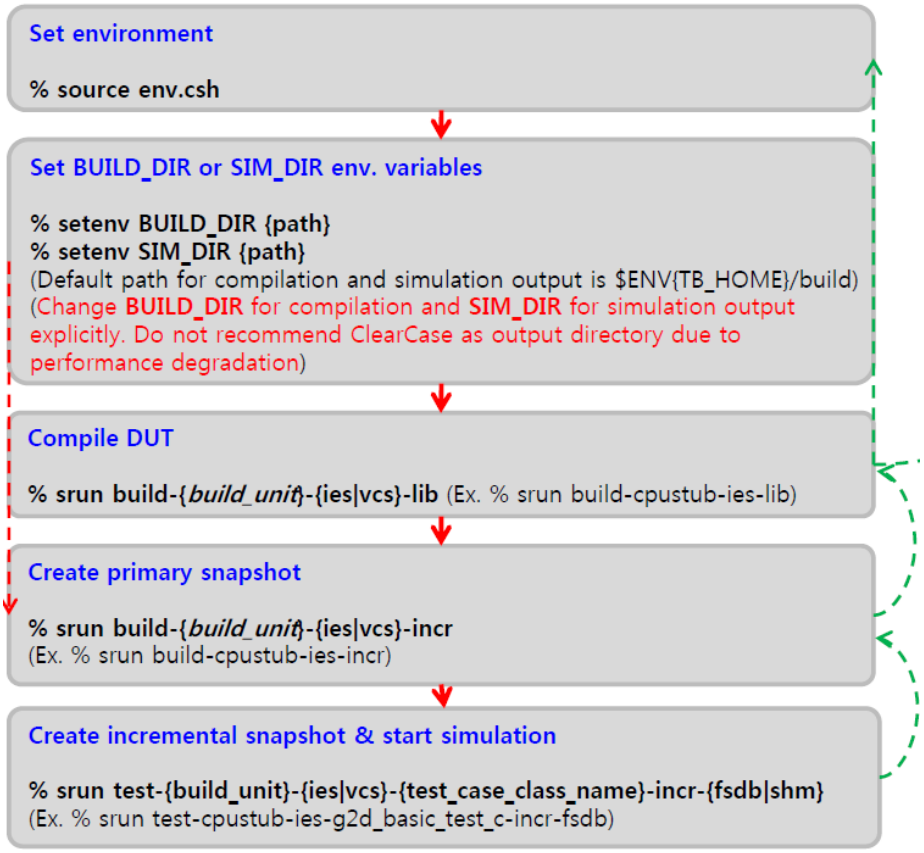

Figure 3 SIVPD simulation flow

SIVPD also encouraged us to perform apple-to-apple comparison of the various simulators with similar DUT and TB. A plethora of methodology improvement features and performance knobs were instilled into the deployed SIVPD for our SoC which further improved the run-times and overall target of achieving faster verification turnaround time. These included fruitful experiments with save and restart, hierarchical reference permissions, read/write/connectivity debug access permissions, simulator performance switches, wave dump options, LSF and compute memory optimization et al which will be discussed in detail in final paper. We also faced roadblocks with respect to Verilog and SystemVerilog LRM and UPF compliance design and testbench code which was modified on-the-go as well as tool crashes which was rectified. Two aspects which surprised us were deployment of Tool2 as primary power aware tool to avoid a sticky issue from the competition which was the signoff tool and the runtimes being better on Tool2 compared to the competition. This resulted in us driving Accelerated SIVPD (ASIVPD) which resulted simulator speed improvements upto 7 times on RTL and 9 Times on power aware simulations (Table 1). The SIVPD flow is now deployed on a multi mullion gate SoC (14 complex subsystems with cross interaction) with Tool2 as signoff for the first time.

Type Of Sim/Vendor Run Time (Seconds)	Tool1	Tool2	Tool2 Vs Tool1
RTL Sim	29501	7305	4.04
RTL Sim With Dump	62145	9207	6.75
PA RTL Sim	63405	10807	5.87
PA RTL Sim With Dump	130005	14550	8.94

Table 1: Tool1 vs Tool2 Run Times

Future scope trials runs are in final stages for optimization of all vendor simulation times. In depth simulation times analyses on various runs will be published in the main paper. We are also evaluating on piloting SIVPD for emulation and FPGA platforms.