2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Analog Mixed Signal Verification and Validation(V&V) Methodology: Bridging the Gap between Pre Silicon Verification and Post Silicon Validation

Vidya Timmanagoudar, Marcel Oosterhuis, Frank Klosters, Steef Grimbergen, Aravind Rajashekhar

(vidya.t@nxp.com , marcel.oosterhuis@nxp.com, frank.klosters@nxp.com, steef.grimbergen@nxp.com, aravind.rajashekhar@nxp.com)

Abstract- **This paper presents a proven Analog Mixed Signal (AMS) Verification and Validation (V&V) methodology, bridging the gap between pre silicon verification and post silicon validation, focusing on the functional testing aspect of post silicon validation. An idea to reuse the verification testcases in post silicon functional validation led to this methodology. Successfully implemented across many projects, verified and validated several products, with one day bring up of silicon.**

Keywords*— Analog Mixed Signal (AMS); Verification and Validation (V&V)*; Simulation;DUT

I. INTRODUCTION

In most cases the pre silicon testcases cannot be re-used in post silicon validation, which leads to more effort and time in post silicon validation**.** This paper presents a proven Analog Mixed Signal (AMS) Verification and Validation (V&V) methodology, bridging this gap effectively. It focuses on the functional testing aspect of post silicon validation**.** The verification environment is based on the "Validation" concept used in the silicon validation. This concept assures reusability of the simulated tests in the silicon validation. The stimuli timing in the silicon validation is exactly the same as in the simulation, making it easy to visualize the validation signals waves in the same picture, next each other with the simulated waves. Section I describes the SETUP (validation and verification), Section II describes the METHODOLOGY, Section III describes the RESULTS and CONCLUSION and Section VI describes the FUTURE IMPROVEMENTS

II. SETUP

A. Validation Setup

The validation setup consists of Main board, DUT board, USB oscilloscopes, PC and Power supplies as shown in Figure2. Central brain is the FPGA. The sequencer (IP in the FPGA) transfer's vector bits to and from the IO's, which forms the central nervous system. Currently running at 50MHz, giving 20ns resolution. Support logic on PCB are: SDCARD interface to hold Linux O.S. and FPGA image, Ethernet port to communicate with PC,USB (uart) act as terminal port for Linux . Board SMPS generates all needed supplies from 35V supply input, enabled via sequencer. It also supports programmable supplies, digital IO's, pull up/down, remote CAN/LIN transceivers, trigger

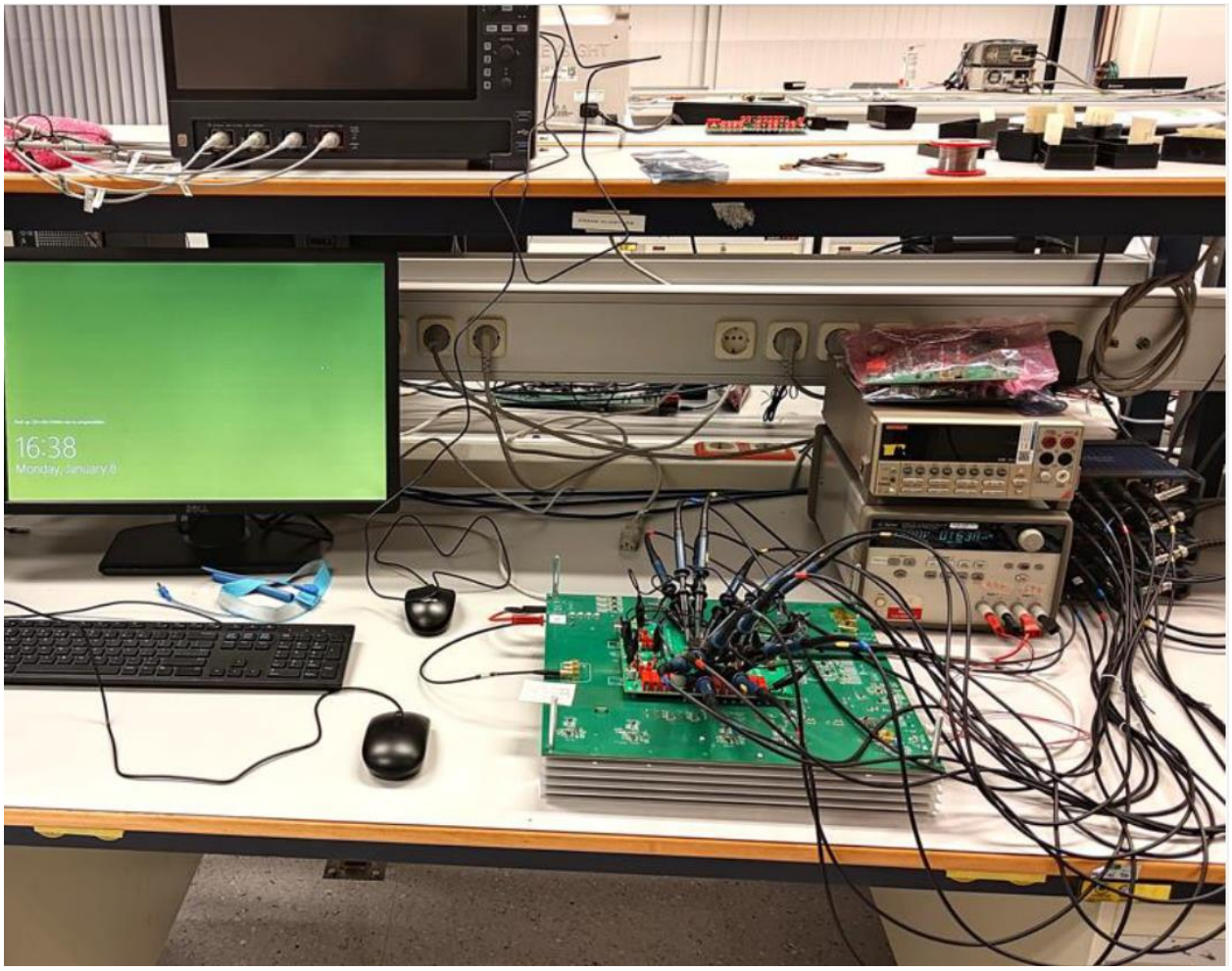

Figure1: Validation setup in the lab

Main goal is to validate the SOC requirements and functional behavior of the device through high-level use cases. This setup generates defined stimuli and captures the device's response by using an oscilloscope, this is based on re-use of the verification tests. Checks on the device's response are compared and a pass/fail conclusion can be made

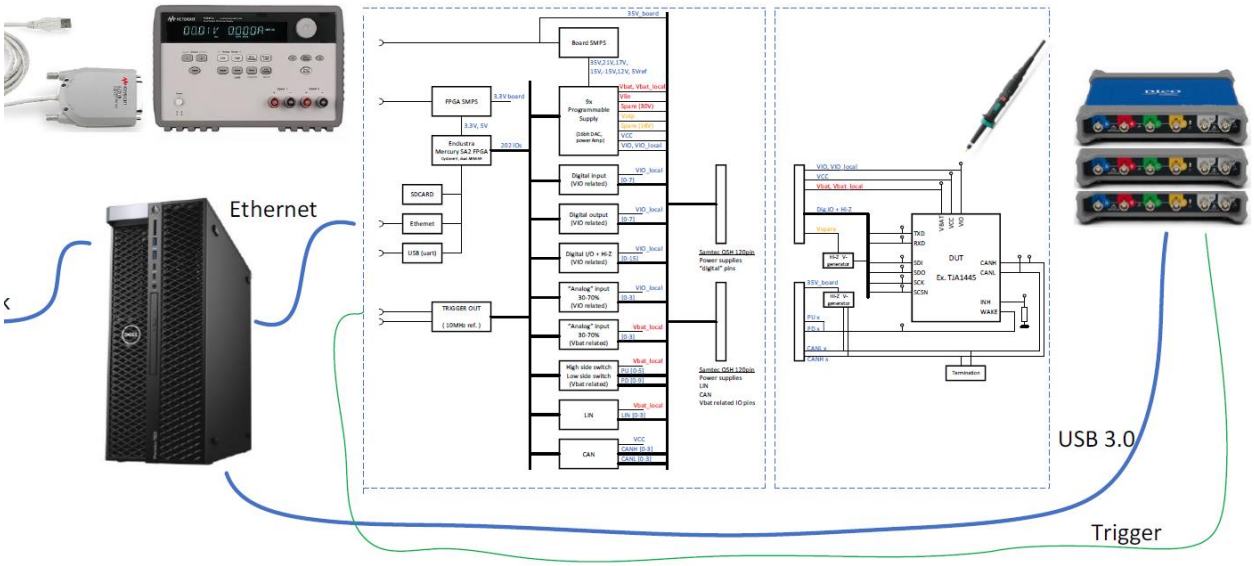

Figure2: Overview of the Validation setup

B. *Verification Setup*

The verification environment models the validation platform as in Fig 3 and 4. Verification testbench mimics the components in the validation setup like the IO drivers, DAC's etc. Testcases are written in python jupyter notebooks .The intention is to have all necessary information and the result for every test in one single document, the jupyter notebook. It uses a multi variant environment approach where a single project verifies a family of products.

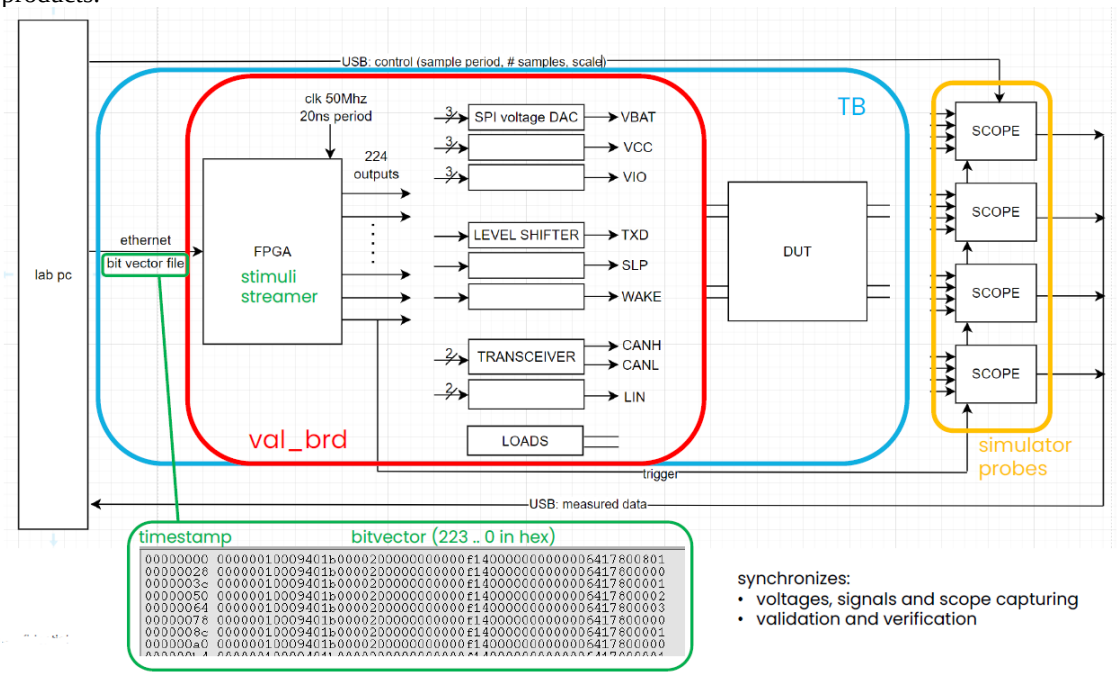

Figure3: Verification setup

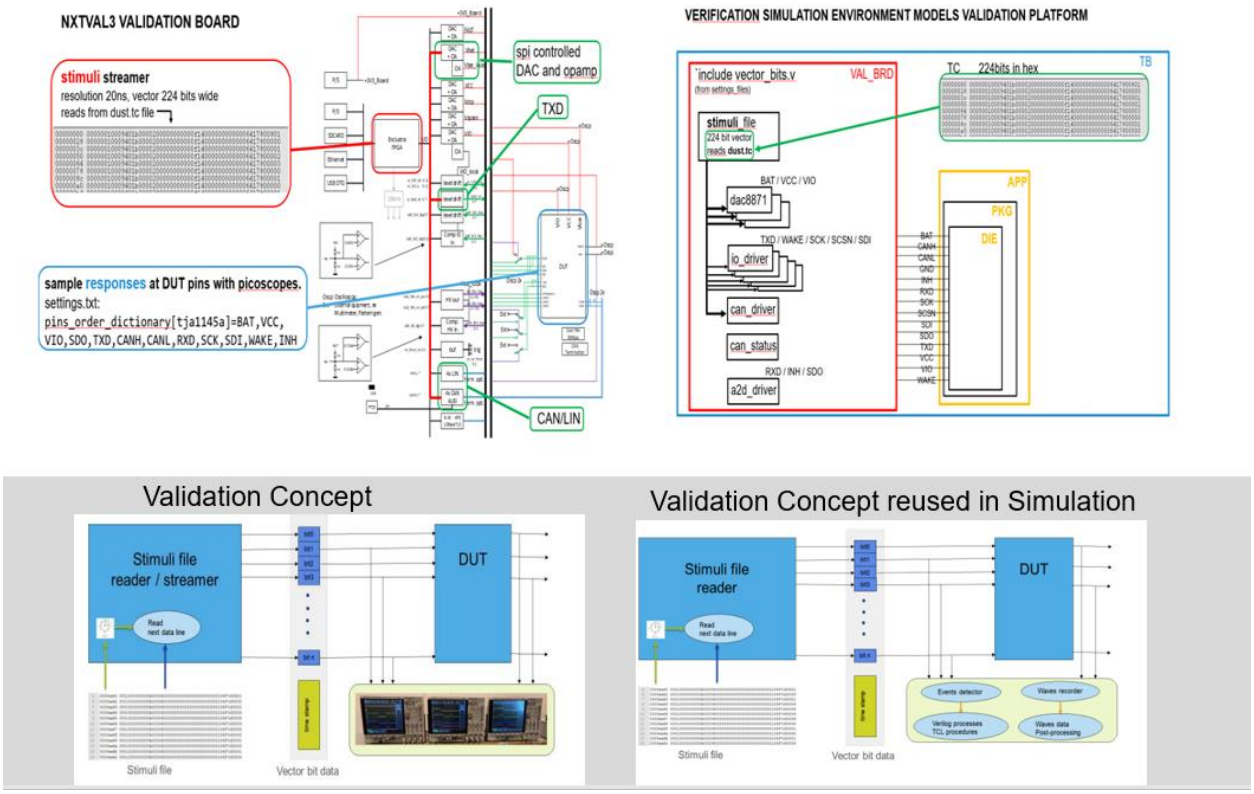

Figure4: Vérification environnement models Validation Platform

II. METHODOLOGY

All tests are written and documented in jupyter notebooks.The test source code is converted into “stimuli” format and sent to the DUT as a burst at the times indicated in the time stamps, first column of the “stimuli” format. The second column represents the vector bit data, every bit in the vector bit data is mapped to a DUT input signal or to special signals which needs to be generated during the test executions. In silicon validation, this process is implemented in the FPGA and in the simulation environment this is implemented in a simple Verilog file. In verification the simulator runs the simulation and results are stored in the form of waves and csv. In validation the results obtained in vcd are converted to waves format and transferred to linux for comparison. Because the simulated wave has the same timing and the same color as the validation wave it is very easy to detect and analyze the differences. “Waves post processing” - running after the simulation has been finished or the data has been captured in the validation, analyzing the entire data generated during the test. The checks development or debugging does not require a test to be re executed, which can spare a substantial time in case of schematic simulations.

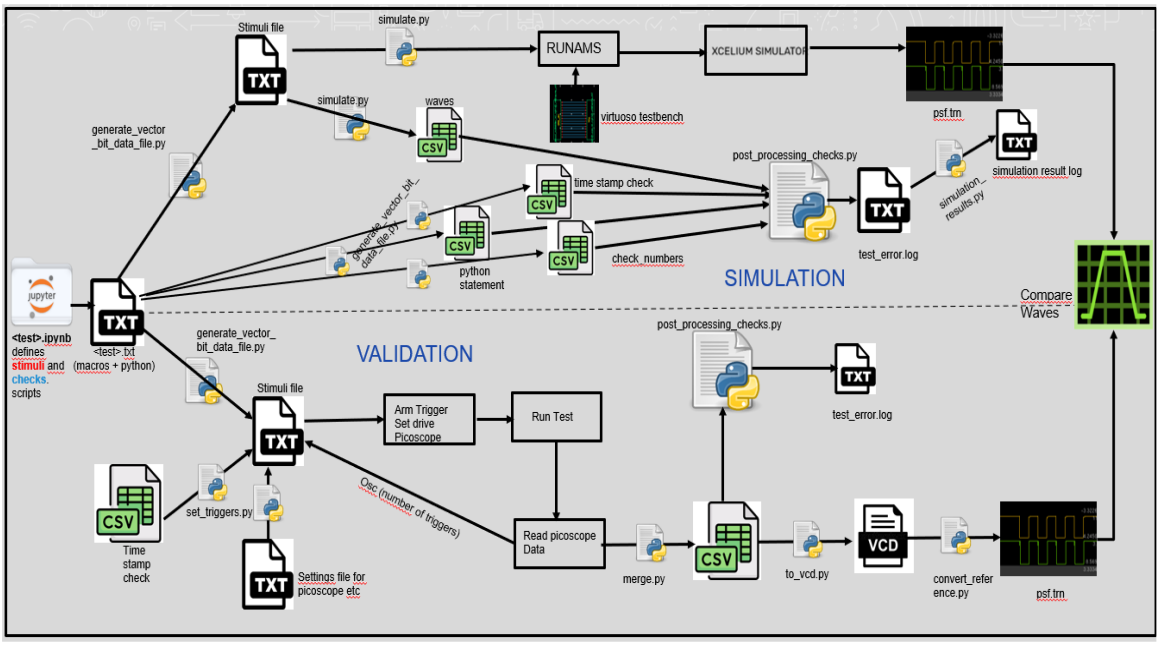

Figure5: Flow Chart of the Methodology

III. Results and Conclusion

Figure6 below shows waveform comparison between verification and validation. Proven methodology, used across projects and several products. Effective pre silicon verification of many variants of a product. Greatly accelerated functional testing in post silicon validation. Overall it proved to bridge the gap between pre silicon verification and post silicon validation by re-using the pre silicon testcases to accelerate and reduce effort during the functional testing of post silicon and also improves the post silicon validation coverage .

IV. Future Improvements

Further, plan to deploy this methodology by unifying it and making it more streamlined. Research on new features to use machine learning, AI, cocotb(built in subroutines, standardized approach).

Figure6: Simulation waves compared with Validation waves (discrete)

V. ACKNOWLEDGMENT

Thanks to Neculai Ailenei, Gerald Kwakernaat, Henri Vos, Jan Vink, Gert-Jan Tromp, Frank Klosters, Marcel Oosterhuis, Aravind Rajashekhar for their contributions

VI. REFERENCES

[1] Validation first flow used for AMS products – Frank Klosters, Marcel Oosterhuis