Advanced Functional Verification Methodology Using UVM for complex DSP Algorithms in Mixed Signal RF SoCs

Srinivas Aluri
Texas Instruments, Dallas, TX

Jaimin Mehta
Texas Instruments, Dallas, TX

Abstract - Complex mixed signal SOC's involving RF front end and signal processing core at base band create a unique challenge for functional verification. Commercial tool like matlab provides powerful built in functions, to realize DSP functionality, stimulus generation and power spectral analysis. RF functionality is realized using analog modeling with wreals. End-to-end golden reference model is created using block level models which allows re-use and greatly reduce verification cycle time. Metric driven verification methodology using UVM on the other hand gives this approach a whole new dimension by bringing in power of constraint randomization, mixed Signal assertions and coverage [1], to completely verify the mixed signal state space. Literature study shows that previous work in this arena of integrating signal processing tools with HVL, required additional overhead of knowing tool API [2]. This approach also has a bottle neck of availability of tool licenses specially when running huge number of regressions. We present seamless integration of above mentioned concepts without the knowledge and coding overhead of API, to effectively verify mixed signal SOC's by creating a executable of reference model/checkers and calling it from HVL.

Key words: UVM, UVC, SV, Correlation, SOC

1. INTRODUCTION

Mixed signal SOC's specifically those with significant analog and digital Content offer unique challenges in design verification. In addition to the state space, Digital Signal Path adds a different dimension to the complexity. In this paper, we present a methodology to integrate commercial signal processing tool into HVL to create a robust self-checking test bench which can thoroughly verify complex DSP algorithms.

2. Verification Architecture

DUT in this work is an RF transceiver SOC with integrated MPU to configure all analog and digital components of the design. In this paper we illustrate two scenarios a) verification of receiver signal processing chain b) algorithmic verification of correlation module, where mat lab integration into UVM proved to be extremely useful.

Fig.1 Shows UVM test bench architecture with various UVC's for configuration of signal processing chain and stimulus generation. Analog frontend and mixed signal modules which transfer data to DSP are implemented in discrete domain using VerilogAMS to speed up simulations. SV interfaces and monitors are used along the signal chain to collect data at the input and output of each component and transfer data at transaction level into the scoreboard.

Fig.2 shows UVM test bench architecture where in reference model is written in matlab and is integrated into UVM test bench.

3. Matlab Integration into HVL

Building a robust self-checking test bench and effective re-use of system level models to reduce design cycle time are driving factors to integrate signal processing models into UVM. In this work we present a methodology of calling matlab from systemverilog which does not

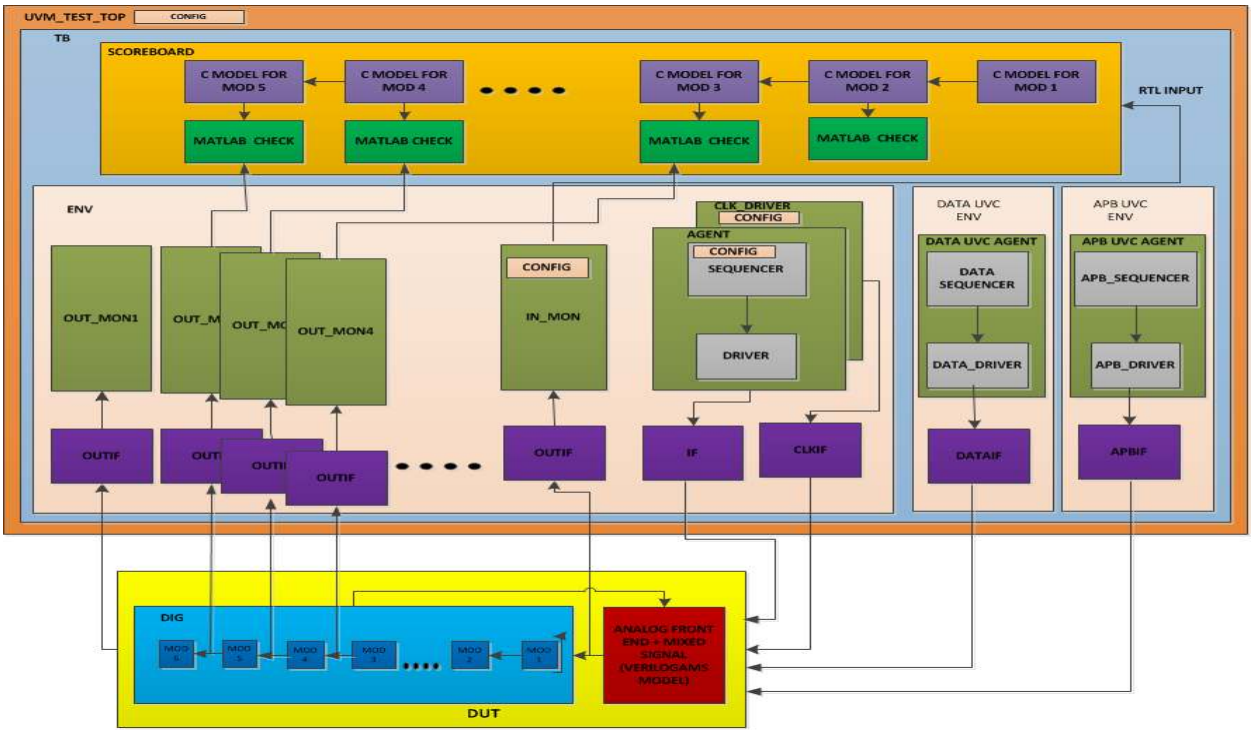

Fig1. Self-checking UVM test bench architecture, showing Integration of reference model in c and checkers in matlab.

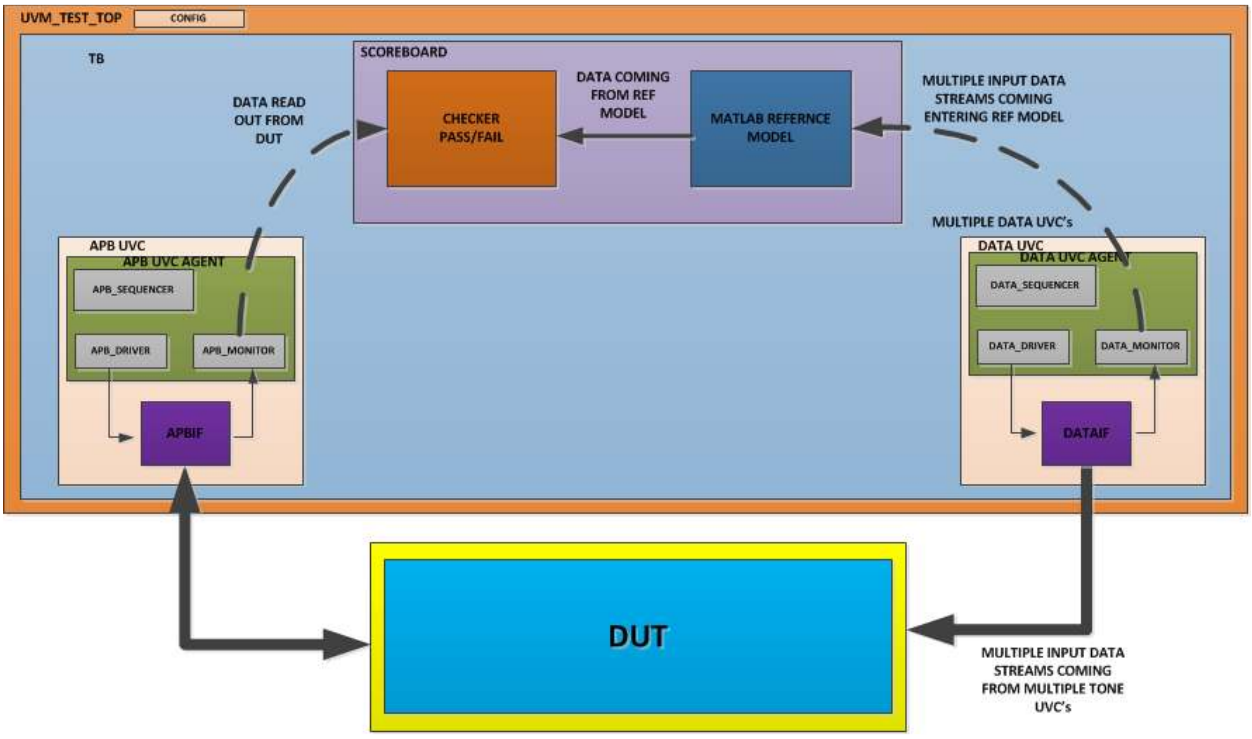

Fig2. Test bench architecture showing matlab reference model integrated into UVM

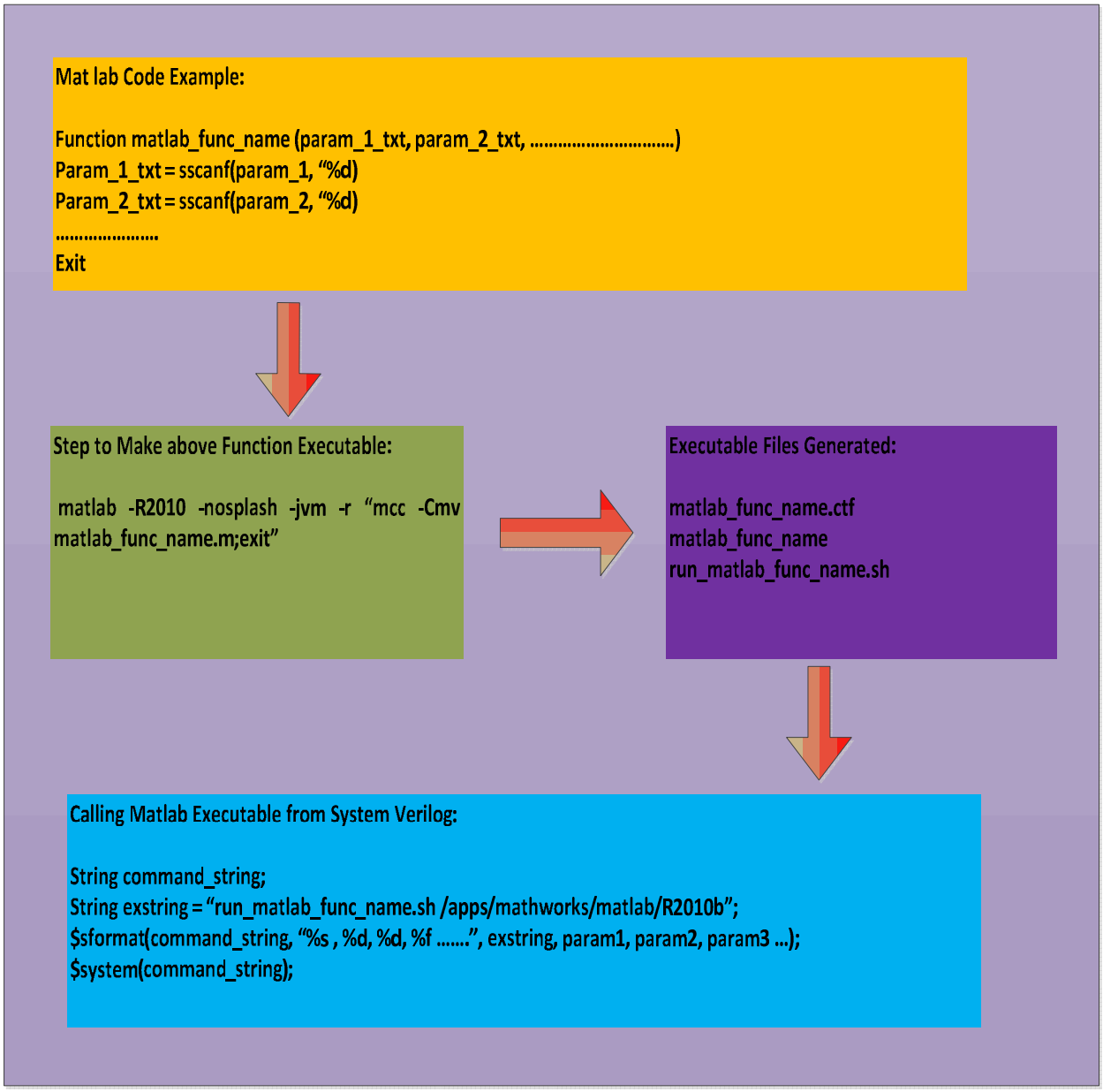

Fig3. Flow chart showing integration of matlab with HVL

require any API knowledge. Golden reference model/checkers which are realized as matlab functions are first converted to standalone executables using matlab compiler. Fig 3 shows detailed Flow chart illustrating the integration.

4. Summary

The above mentioned procedure was successfully used to call complex matlab signal processing functions and reference model created in mat lab. This methodology helped us in creating effective self-checking test bench architecture, and enabled us to re-use most of the system level functions and reference model thus reducing verification cycle time. Complex signal processing functions can easily be coded in matlab and called from systemverilog, which reduces the complexity and many lines of code needed if the same is implemented in HVL's. Finally this methodology also has proven effective when running lot of regressions, since it does not need to invoke the matlab engine, thus does need many mat lab licenses while invoking from HVL's.

5. References

[1] Neyaz Khan & Yaron Kashai, "*From Spec to Verification Closure: a case study of applying UVM-MS for first pass success to a complex Mixed-Signal SoC design*", *DVCON 2012*

[2] Dhaval Modi, Harsh Sitapara, Rahul Shah, Ekata Mehul, Pinal Engineer, "*Integrating MATLAB with Verification HDLs for Functional Verification of Image and Video Processing ASIC", International Journal of Computer Science & Emerging Technologies, Vol 2, Issue 2, April 2011.*