2015 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

A concept for expanding a UVM testbench to the analog-centric toplevel

Felix Assmann, Bosch Sensortec, Dresden, Germany (*Felix.Assmann@bosch-sensortec.com*)

Axel Strobel, Bosch Sensortec, Dresden, Germany (*Axel.Strobel@bosch-sensortec.com*)

Hans Zander, Cadence Design Systems, Feldkirchen, Germany (*hans@cadence.com*)

***Abstract* - This paper outlines a new approach to improve the productivity in mixed-signal verification by addressing both analog and digital verification requirements in one testbench. The UVM verification environment which was used on the digital portion of the design was enhanced to address the needs of the analog verification as well as the full chip level verification. The paper describes the concept, the actual architecture, and the implementation of the new UVM-mixed-signal testbench followed by some use case examples and the results and experience made with the new approach.**

Keywords—UVM; MDV; mixed-signal; wreal; testbench-reuse; whitebox; blackbox

I. Introduction

Typical mixed signal designs contain a digital and an analog portion. While in general the complexity of our chips – especially of the digital part – is growing, the designs are still analog centric. The verification of these mixed-signal designs has always been challenging as there was no common approach for digital and analog verification. There are already existing approaches to deal with analog centric designs with real number modeling [1, 2]. But especially with the introduction of advanced verification methodologies, there is an even bigger gap between the digital verification – which is based on metric-driven verification (MDV) and the Universal Verification Methodology (UVM) – and the analog verification – which is rather directed and use case oriented, with manual inspection using a waveform.

The growing complexity of the designs also has an impact on the simulation performance and the verification throughput. Especially with the increased number of simulations due to the randomness of the UVM testbench it is not feasible anymore to run all regression tests at the full chip level.

However, the verification of the full chip is mandatory to identify bugs in the analog/digital interface, which can only be found on toplevel, as well as full chip behavior. Today common approaches distinguish between analog and digital verification on toplevel [3, 4]. Such that the verification cannot benefit from analog designers knowledge of the analog part. While it is not realistic to port the ways of working from digital to analog and visa-verse, a new concept needs to be found. It should meet the requirements of fast simulations with appropriate accuracy and should still enable different flows of verification in order to leave both domain experts in their comfort zones.

II. Challenges

Even though we have already been using wreal models to be able to increase the simulation performance, the verification concept was not aligned yet. The verification activities of the analog and the digital team were quite disconnected and based on the usage of two different testbenches:

A. *UVM based testbench along with wreal models*

- Used for digital only simulations and full chip simulations using wreal models
- Not compatible with transistor level simulations

B. *TCL based testbench along with analog schematics*

- Used for scenarios which required full precision (e.g. power-up, power mode transitions, functionality with strong interactions between digital and analog)
- No automatic checks available

- Heavily used because UVM environment was often under construction and was thus sometimes blocking (e.g. due to compilation errors) the verification progress on the analog side

Maintaining these two testbenches required a lot of effort. The hardest part was to always ensure that the behavior of both testbenches was consistent. We observed different behavior of the wreal models based on simulation parameters, for example delay_mode_zero together with delayed assignments:

$$assign\ \#3\ a = b; \tag{1}$$

Equation (1) does not delay the signal assignment, but wreal models relied on this kind of structure for e.g. enable delay modeling of analog blocks. Sometimes it took a huge amount of time and long debug cycles to identify such issues.

Finally, the scripting environment was not flexible enough. In the analog domain the functionality of a config view is heavily used. It allows a substitution of schematic level with a wreal model and vice versa inside of the same testbench. The old approach did not easily allow to use the UVM environment with any specific analog config view. As a consequence, there was just one working “configuration” which used wreal models only together with the UVM environment. It was very hard to replace selected wreal models with real schematics which could have improved the development of analog modules inside toplevel context. Just changing simulation scripts could not solve the problem of missing encapsulation and configurability.

The challenges stated above had a big impact on the verification productivity and brought a high risk for critical bugs to escape. In order to overcome these challenges, we developed a concept for expanding the UVM testbench also to the analog-centric toplevel.

III. Concept

To address the simulation performance we used the already proven approach of creating abstract wreal models for the analog circuits. To serve the overall concept, this time it was required to create several models for the analog blocks, all with different abstraction levels. As also third party models were used, some of them had to be remodeled or enhanced to provide either higher precision or higher performance with reduced precision. In general we needed at least two models of each block with different accuracy. For example a supply ramp only needs to be accurate enough to trigger power-on reset thresholds properly for digital verification

The associated question of verifying the functional equivalence of the model and the circuit will not be discussed here.

Creating a common and configurable testbench was the more challenging task. As stated above, it is not feasible or even desired to fully port the verification methodologies and tools from the digital/analog world to the other. The main focus was to create a common testbench which allowed both digital and analog verification without any unnecessary dependencies.

A. Partitioning of the UVM verification environment

For the digital part we used a bottom up approach using UVM environments on block level, reusing them up to the full digital toplevel. Having mixed-signal verification in mind, two derivatives of the toplevel verification environment were required:

- BlackBox environment
  - Used for testcase development and toplevel development (analog and digital)
  - Always working, contains only interface components (interface UVCs)
    - Only functionalities which an external master would execute from the chip boundary were integrated
    - Does not contain any checking components
  - Used by analog designers and verification engineers for directed tests with waveform analysis
  - Interactive command interface to e.g. program registers or stimulate pins via generic pin drivers
  - Checks are specified inside of testcases or as part of the interface UVCs

  - Also used for gate level verification
- WhiteBox environment
  - Used for regression
  - Contains all block level UVCs (internal passive UVCs, SBs, internal checkers, etc.)
  - Imports BlackBox environment for modular reuse
  - Used mainly by verification engineers
  - All tests from the BlackBox can be run in the WhiteBox
  - Usable for digital part as RTL description only
  - All BlackBox features (e.g. interactive mode) are usable

B. *Digital verification*

Classical UVM/MDV approach including interactions with analog part via wreal model

C. *Analog verification*

The verification of the analog blocks was done by each analog designer in the way described in the introduction. Once verified they were contributed to the toplevel environment.

In addition to that, the new testbench concept allows the adoption of full custom logic for analog modules within the toplevel design. Because control signals and methods to drive them are already in place (e.g. just set a register via the interactive command interface), the analog designer can focus on the development of his/her module. With the new setup there is no need to create a stimulus model (which could be different for each analog block).

To further enhance the checking of the analog part (e.g. trimming, offsets) and to be able to model impurities of the analog blocks, configurable wreal models were added. They can be configured during run-time via an active UVC which drives and reads a parameter file (cds_globals) for the analog wreal models (see Figure 1). Furthermore this gives us the opportunity to add randomness to the analog part.

D. *Top level verification*

The toplevel testbench contains either the WhiteBox or the BlackBox verification environment. For analog verification the BlackBox environment is chosen for interactive debugging and development. For regression verification, mainly the WhiteBox environment was used together with tests which were developed in the BlackBox environment. It is also possible to use the BlackBox environment as part of the regression (e.g. gatelevel verification), but then the tests need to contain the checks.

To be able to also drive the analog ports, generic pin drivers were grouped under a so called stim-core. The stim-core represents the interface of a virtual master and is driven by an active UVC (e.g. SPI-VIP) from the UVM verification environment. The full testbench architecture is illustrated in see Figure 1.

A command interface was created to also enable the analog designers to control the UVM verification environment during simulation (e.g. register writes). Only the given command is executed and simulated, which gives the designers full interactive capabilities.

IV. Architecture & Configurations

To be independent of the signal representations (logic signals on digital, any analog port representation analog, etc.) a generic pin driver component was created. Pin driver modules replace the previously used simple connect modules which were not functional for analog simulations because of the digital outputs, missing pull-ups, and slew rate settings. The new approach models the functionality of master device pad stages (the testbench models the reality in the sense of master slave communication). Furthermore the models were extended by a clock generator to allow a simulation efficient method to apply test clocks to any ASIC pin, e.g. to simulate scanmode. The architecture of a generic pin driver module is illustrated in Figure 2.

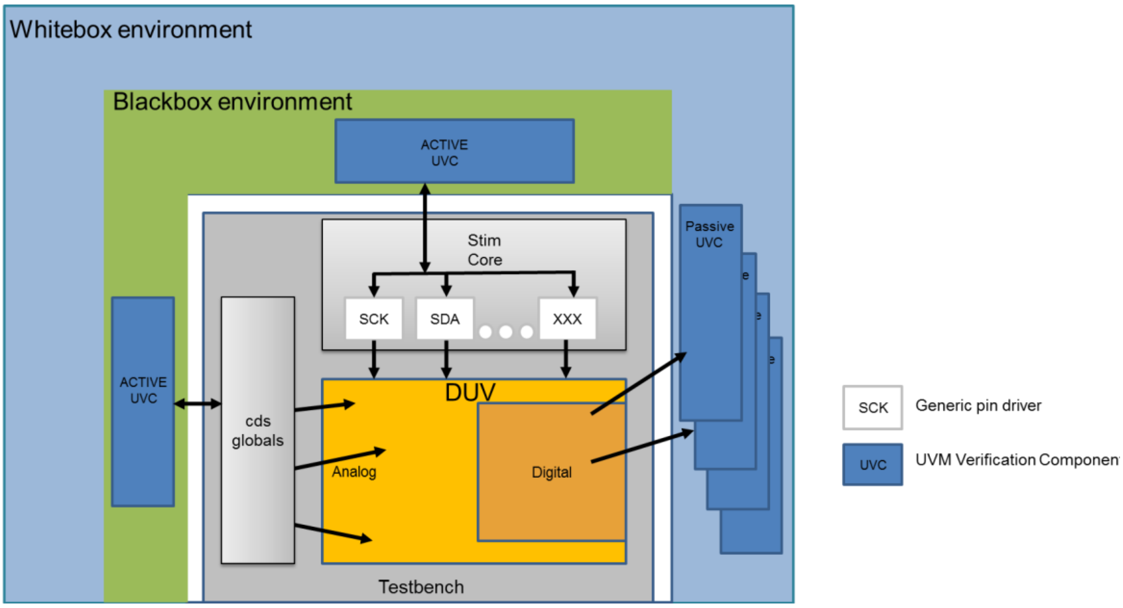
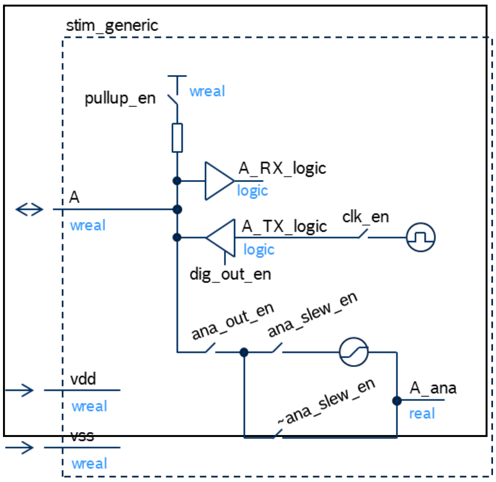


Figure 1. Full testbench architecture



Driving scenarios:

- A_tx_logic for the digital value
- Dig_out_en to enable digital driver
- A_ana for supplys
- Ana_out_en to enable analog driver
- Ana_slew for supply ramps

Figure 2. Architecture of a generic pin-driver

A. Configuration & Usage examples

The TB is used in different regression configurations. Each configuration serves a specific purpose. For example all digital centric configurations only use the digital simulator to be faster. But there are also configurations which use the analog solver if the precision is needed. Below are some configuration examples:

- Digital config A
  - DUT configuration
    - All analog modules as wreal models (high precision models)
    - All digital modules as RTL
  - Scenarios
    - Power up sequence
    - All test with interaction to external system components
    - System level test sequences
- Digital config B

    - DUT configuration
        - All analog modules as wreal models (low precision models)
        - All digital modules as RTL
        - Simple third party / IP block wreal models
    - Scenarios
        - Long running sequences
        - Sequences with focus on the digital side
- Analog config A
    - DUT configuration
        - All analog modules on transistor level
        - All digital modules as RTL
    - Scenarios
        - Basic functionality tests
        - Power-up sequence tests
        - Analog/digital signal checks (enable and reset polarity, current direction)
- Analog config B
    - DUT configuration
        - Some analog modules on transistor level
            - Pads and power modules on schematic level
            - Rest wreal
        - All digital modules as RTL
    - Scenarios
        - Power mode transitions

All configurations support both BlackBox and WhiteBox testbenches and can be run in full regression setup.

V. Experience & Results

The new concept provides significant improvements in many aspects. It clearly increased the verification productivity:

A. Only one toplevel testbench

- Only one environment to maintain
- No more inconsistencies
- Partitioning of UVM environment into BlackBox and WhiteBox allows independent and parallel work mode for analog centric and digital centric verification

B. Efficient testcase creation

- Analog designers can do full schematic verification in combination with existing testcases, no rewriting needed
- Directed testcase from analog or digital toplevel bring-up is used as base test for MDV. Only some randomness had to be added

- Analog testcases can now be easily integrated in the regression as they are based on the UVM environment (They only have to be self-checking)
- Single testcase setup and debug about 5x faster due to simulation time enhancements from model configuration

C. *Efficient analog component adjustment*

- Analog designers can easily verify the interaction of the analog circuits together with the real digital input stimuli. Previously many "hacks" had to be done to get this working (with limitations)

D. *Increased verification productivity*

- Previously the verification coverage was limited by long simulation times and TB maintenance issues (wait until change was fixed everywhere)
- Different config views allow different configuration for fast, accurate or other simulations
- Simulation time speedup because of different config views
    - Power up cycle: 10s in digital config B vs. 5min in digital config A
    - A lot of interactions between digital and analog (e.g. control loops) are now simulatable in a reasonable timeframe
        - Fast simulation possible due to wreal models
        - Lots of tests possible due to random UVM environment
- Possibility to verify single wreal models in toplevel context with toplevel/real stimuli

VI. Summary

The new concept was successfully applied on our current project. It proved to increase our verification productivity dramatically. Also the verification coverage increased in all aspects, focusing on functionalities which could not be verified easily in the past. Overall the new concept improved the quality of the product.

New projects will certainly apply this concept, where some enhancements are planned already:

A. *Model checks inside of regression*

- Compare schematic and wreal model in regression (due to missing equivalence checking)
- The checking part is still under discussion

B. *Enhance interactive mode for more granular usage, e.g. control of single register fields*

VII. References

[1] Ira Miller, Ana Ferreira-Noullet, Behavioral Modeling in Industrial IC Design Experiences and Observations, BMAS, 2004.

[2] Jones,K., Analog and mixed signal verification, FMCAD, 2008.

[3] Mixed-Signal System-on-a-Chip (SoC) Verification Based on SystemVerilog Model, 45th Southeastern Symposium on System Theory Baylor University, 2013.

[4] Chao Liang, Mixed-Signal Verification Methods for Multi-Power Mixed-Signal System-on-Chip (SoC) Design, ASICON, 2013.