

Unified Verification Methodology for Digital and Analog Mixed-Signal Co-Simulation for Low Power Embedded SoC

Ashwini Padoor
 Connected MCU,
 Texas Instruments (India) Pvt. Ltd.
 Bengaluru, 560 093, India
 ashwini.padoor@ti.com

Lakshmanan Balasubramanian
 Connected MCU,
 Texas Instruments (India) Pvt. Ltd.
 Bengaluru, 560 093, India
 lakshmanan@ti.com

Abstract— Analog, power management and RF integration in a system on chip (SoC) require extensive focus on verification of analog design, system integration, low power intent, and use-case scenarios to ensure quality and timely product delivery. In this paper, we propose a unified environment and methodology for seamless verification of the mixed-signal SoC focussing digital, analog and low power aspects. The key features involve System Verilog (SV) Universal Verification Methodology (UVM) compliant architecture, reusable test bench and test case structure, common assertions and checkers.

Keywords— Analog mixed-signal co-simulation, AMS, digital mixed-signal co-simulation, DMS

I. INTRODUCTION

Explosion of portable, battery operated, autonomous embedded internet-of-things (IOT) market and related application require low power and low cost as the DNA for all underlying building blocks. This mandates convergence and integration of analog mixed-signal (AMS) contents, power management, and RF [1][2][3][4]. This situation demands smart verification approach and simulation flow for thorough verification targeting AMS integration correctness, analog-digital boundary handshake and timing correctness, system functionality in various power and performance modes, and analog component functional integrity among other basic verification needs [5][6][7][8].

The paper is organized into four sections. Section II describes the unified verification environment. Section III details the reusable test bench and test case architecture. Section IV details coverage metrics and quantification. Section V details common assertions and checkers strategy. Section VI describes in detail the unified environment for DMS and AMS co-simulation. Section VII highlights the hardware software synchronisation methodology. An application of the methodology on an industrial case study is illustrated in section VIII, while the results are discussed in section IX. Section X concludes the paper.

II. UNIFIED VERIFICATION ENVIRONMENT

The shrinking development cycles define the need for a comprehensive, efficient and flexible test bench (TB) platform for the SoC verification. Traditionally the analog and digital features of the SoC have been considered as two different domains and hence the verification approach. Predominantly digital focussed functional verification uses the advanced methodologies and always running checks to catch complex to simple design bugs. Similarly there are various methodologies to verify AMS contents, but they lack in ensuring acceptable, say 99.99%, functional correctness of design. This frequently results in re-spins of AMS design/SoCs for functional errors. This leads to loss of time to market for MS chips resulting in non-profitable ventures. Hence to avoid this type of traditional loop holes in the SoC verification approach we used single verification environment. The AMS features as well as SoC features are verified using the same TB setup as shown in Figure 1,

which has supported all the UVM components (UVC monitors, checkers, and coverage definitions) [5].

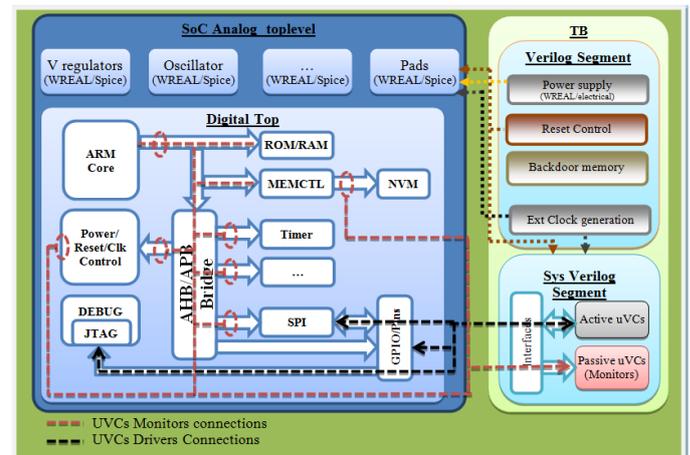


Figure 1. Test bench setup

The unified verification environment enables the user to utilize the same verification environment for functional verification and AMS co-simulation scenario seamlessly. The verification environment is architected in such a way that the user can utilize the flexibility of using real number (RN) analog behavioral (Verilog AMS *wreal*) models, transistor level SPICE representation or a mix of the two for any analog component. This configuration could be selected for each test case (TC) based on the requirements.

This provides the flexibility to use real number (RN) behavioral models (BMOD) for the faster bring-up including associated debug and to reuse the environment including the test cases for the AMS co-simulation just by configuring all or some of the modules to use actual SPICE netlist as illustrated in Figure 2.

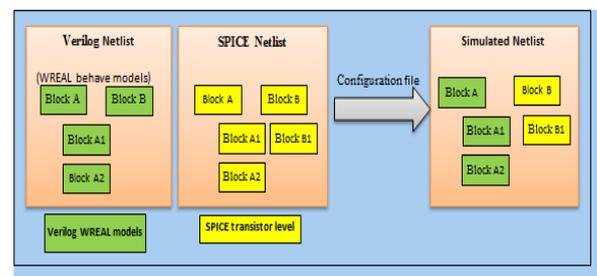


Figure 2. AMS co-simulation configuration

At SoC level, therefore this mixed-signal (MS) verification flow, using RN models and an assertion-based verification approach, brings together the analog and digital sides seamlessly and coherently. Integrating analog behavioral models, analog and digital solvers into one flow, the methodology helps to balance the right amount of accuracy and speed based on the design requirements. The flow delivers:

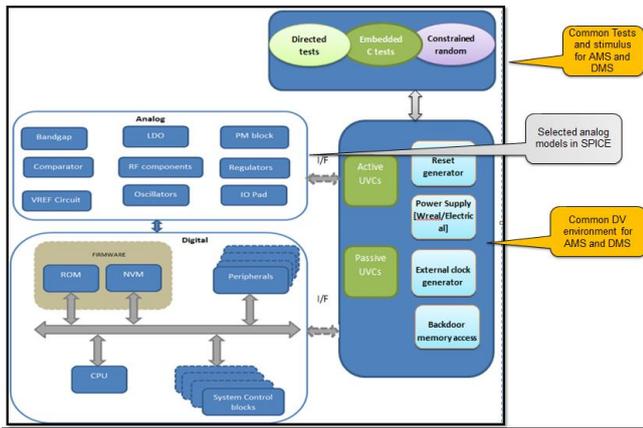


Figure 3. Test bench and test case architecture

- A single verification environment combining both analog and digital verification requirement that can be used to functionally verify at the desired level of abstraction using either option or both digital and analog targeting speed and accuracy respectively
- Metric-driven verification (MDV) approach to analog components in a MS design
- Complex checkers that remain always enabled can check for any combination of cross-domain analog and digital sequences
- Power-aware MS verification

Thus in summary a comprehensive verification approach is achieved at SoC level irrespective of analog or digital simulations using this flow.

The key components of the simulation environment are described in the following sections III through VII.

III. TEST BENCH AND TEST CASE ARCHITECTURE

The TB & TC architecture is shown in Figure 3. The tests are developed such that it can generate either directed or constrained random stimulus. Any test case can be either a UVM sequence or may include an embedded software part. The test case pass-fail criterion is determined by evaluating System Verilog errors (from monitors and assertions), and software (C code) errors along with the test bench timer status. This approach has been used for functional, AMS and netlist based simulations as shown in Figure 4.

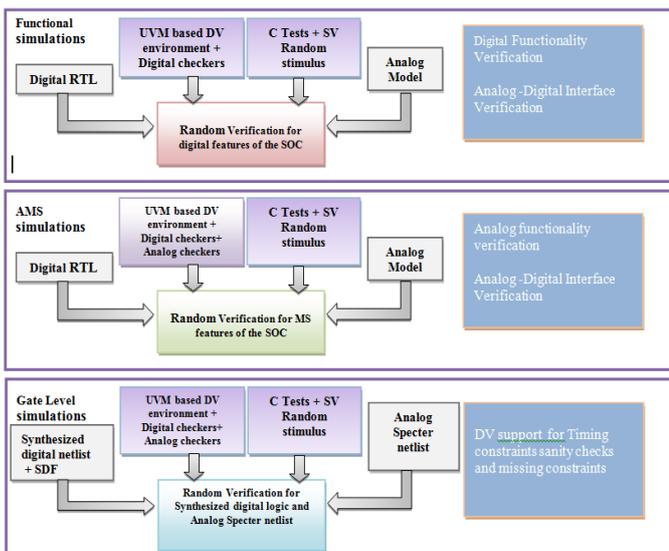


Figure 4. Flow support for different abstraction levels

IV. COVERAGE METRICS

The functional coverage definitions are added to ensure that test cases are meeting the targeted scenarios. The coverage is collected either through coverage groups in the UVC monitors or by assertions.

In order to make sure the IP functionality is covered in detail, coverage groups are defined within the UVC monitors which are IP specific.

Having the possibility to code non-UVM compliant direct tests necessitated an automatic tracking of the coverage for these scenarios. This was achieved by a method known as "Verification Item Tags" (VITag) natively defined in the flow. A unique VITag can be defined for each scenario in the verification plan and should be used for any items that cannot be covered automatically. VITags are implemented in the test bench as System Verilog assertions and are mapped to the verification plan. The VITags are triggered manually either from the C code (by using a library function) or from the test bench (by using a test bench auxiliary task) and can yield to a PASS/FAIL status as illustrated in Figure 5.

```

VITag PASS trigger from software
setVITAG(VITAG_BASIC_SW,RES_PASS);

VITag FAIL trigger from the test bench
tb.setVitag(`VITAG_BASIC_TB,`RES_FAIL);

```

Figure 5. VITag custom coverage handler

V. CHECKERS

Checkers are implemented at various design abstraction levels to ensure that design is clean from all the perspectives. These include:

- Always running assertions & checkers for
 - Analog-digital interface integration checks (Assertions between digital & analog blocks)
 - Clock frequency correctness checks
 - Reset assertion & de-assertion checks
 - System power status checks
 - Glitch checks
- Scoreboard units
 - Memory data correctness checks
 - Register read data correctness checks
- Standard Protocol checkers
 - System bus (APB & AHB) protocol compliance checks
 - JTAG
 - Serial communication interfaces (SPI, UART, I2C)
 - Debug interfaces
- Control and status register checks
 - Test case specific checks based on either C or UVM sequence
- System address decoding checks

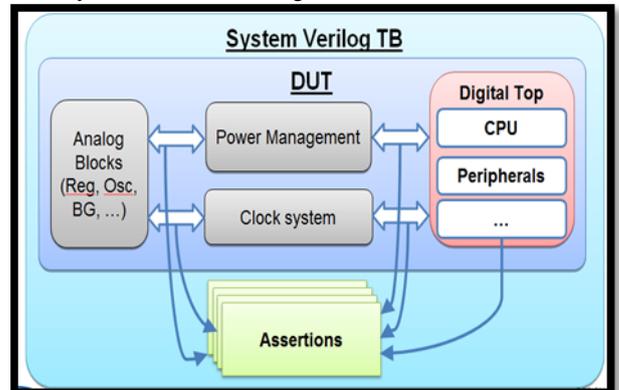


Figure 6. Checker architecture

The checker & assertion architecture implemented is illustrated in Figure 6. These checkers are always enabled and any violations flagged out by providing the appropriate errors details. All the bus protocol checks are enabled for all the interfaces irrespective of the number of instances connected.

Further all analog assertions are developed in such way to have the core analog measurements and observations are done using Verilog AMS and saved to *real* variables. These *real* variables are further used in System Verilog assertions combined with other “digital” signals to have complex checkers. This enables analog assertions to be included in the test case and dashboard pass/fail criteria and coverage metrics in an automated way. There is a small set of analog assertions which were not part of the automated test case or dashboard status monitoring and require a custom automation to observe into the simulation log files.

VI. UNIFIED VERIFICATION ENVIRONMENT FOR DMS AND AMS CO-SIMULATION FLOW (RTL & GL)

The unified verification environment for DMS and AMS co-simulation is illustrated in Figure 7.

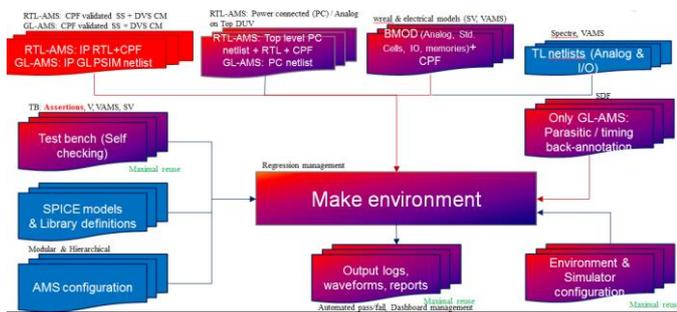


Figure 7. Unified environment for DMS & AMS co-simulation

The SoC under consideration is integrated using the custom analog-on-top (AoT) design integration method. The digital portion represented in RTL is taken through the automated power intent driven low power design flow. The SoC top level has the full physical power connectivity even during early RTL stages of design maturity. A wrapper in schematic is created for the digital portion of the design with power ports, connectivity & supply sensitivity information needed to handle connect modules [7].

Named reference modes are created with set of baseline options defined to avoid redundancy, improve modularity and reuse. Various simulation modes support respectively the CPF based power aware (PA) RTL simulation, non-PA RTL, gate level (GL), PA RTL based AMS co-simulation, and GL AMS co-simulation. In addition there are additional sub-modes to handled appropriate PVT corners both in RTL & GL phases. In GL phases, the match between the SDF timing annotation corner and the PVT corners set for analog portion of the design in SPICE configuration is ensured. Some of the corners for DMS & AMS co-simulation are Nom, Min, Max, Trimmed/Untrimmed (All IPs, BG only), HTOL (Min, Max), & IDDQ.

Mode specific trim & device register configurations is ensured at appropriate device boot-up time in automated way. Analog & AMS configuration is handled in a modular way to avoid errors due to repeated, manual configuration; and to provide flexibility to override the baseline configurations or specify incrementally at test group and test case levels. A baseline configuration applicable for all test cases is defined to automatically take effect. Simulation mode specific configurations and overrides were made possible. While the digital event driven simulator is maintained the same across all modes (Cadence[®] NC simulator), an option called simulation flavours is provided to choose between a more accurate Spectre APS[®] and much faster Spectre XPS MS[®] fast SPICE simulator[9][10]. In addition there are specific modes created to perform some automated custom checks that are useful for low power designs like the dynamic circuit checks using Spectre APS[®] [11]. An additional flexibility is built in the environment to maintain differential simulator or flow version control for various

simulation modes and flavours. This is done to avoid unnecessary dependency between them especially to handle vendor tool specific bugs and workarounds. This is enabled for different simulation flavours including DV/DMS, AMS-APS, AMS-XPS, AMS-APS-DC (dynamic check).

Appropriate RN stimulus is used in the TB to enable seamless migration between DMS/AMS co-simulation. Additional care is taken to ensure low impedance in the supply paths. Further, relatively relaxed accuracy is maintained as a baseline for all analog-digital interface boundaries during AMS co-simulation for the gross functional focus at SoC level verification. Only for select analog paths especially those involve real-electrical transition (either between analog BMOD or TB stimulus/load and SPICE partitions) a higher accuracy is set primarily for more accurate real to electrical translation or vice-versa, as required. For example an RF envelope generator is used to model the free space antenna interface to speed-up AMS co-simulation (only applicable for amplitude modulation) & small signal sinusoid about a DC common-mode level is modelled for external crystal interface [10].

Reusable smart delays are provided in test cases that excite analog components, for ease of automated tracking of design level delays and settling times to have a dependent test case simulation progress. This is to avoid conventional approach using number of reference clock cycles for delay configuration, which can be inaccurate and operate at much higher granularity. Such a conventional method is difficult to handle especially with the system using multiple on/off-chip clock sources involving unrelated setup and settling times.

Additional user options are provided in the environment to perform any individual or group of simulations with debug and profiling options [10]. This is needed to get useful simulation profile and debug information from the analog simulator that enables easy localization of any simulation performance bottlenecks. Assertions are configured for categorized handling and separate handling of analog assertions, where needed.

Various quantities used in analog stimulus, load and programmable configurations in embedded software that affect analog behavior are controlled through appropriate parameters with nominal and acceptable ranges specified. The default behavior is to have the nominal condition for all these; they can be configured to an appropriate minimum or maximum condition as per simulation mode, simulation flavour, test group and test case requirements. This enables a constrained random verification approach targeting analog contents of the design. As an example, some of the parameterized quantities include analog component settling times, voltage or current levels, timing delays, timing constraints, and analog trim configurations.

VII. HARDWARE SOFTWARE SYNCHRONISATION METHODOLOGY

The underlying concept is illustrated in Figure 8.

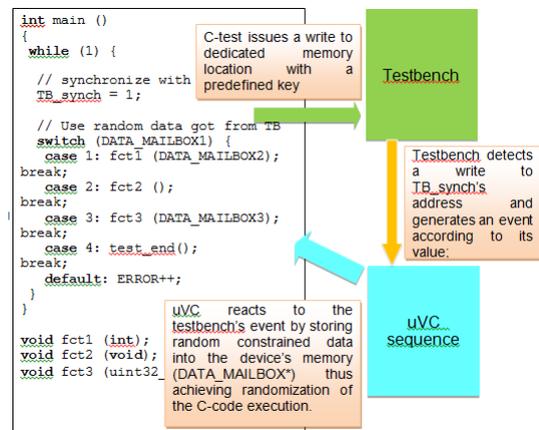


Figure 8. Hardware software synchronization

This is primarily needed to have a seamless interaction and control flow between SV UVM verification environment and the embedded software (C code). One of the key examples where this is very useful is the smart delay application discussed in Section VI for AMS co-simulation to enable automatic delay tracking across PVT variations.

VIII. CASE STUDY

The methodology described in this paper was applied on multiple development execution of complex AMS SoC with analog, power management and RF integration. A high level block diagram of a representative system and a typical AMS co-simulation setup is shown in Figure 9. Rest of this section describes some of the scenarios we covered during the AMS verification using this TB and related observation.

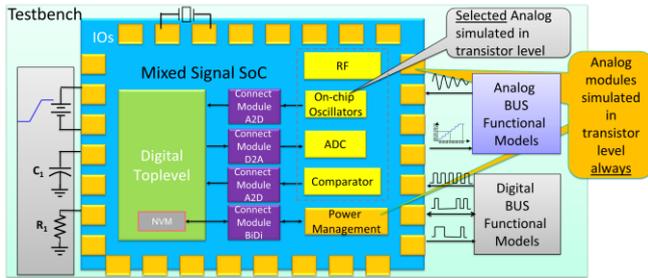


Figure 9. Typical AMS SoC & AMS co-simulation setup

A. NVM IP Test with External Reference

The targeted scenario involves a test scenario of an on-chip supply (LDO) which powers an on-chip non-volatile memory (NVM). During the test scenario, the input reference to the LDO was forced from an external tester source. The said LDO regulator has an option of external voltage reference for testing NVM IP at different voltage corners to ensure appropriate voltage schmo application as NVM supply, in the absence of sufficient output level programmability for the LDO itself. This requires the internal reference to be first disconnected before an external reference is forced through an appropriate analog DFT mechanism. Any inappropriately long delay between switching would cause internal capacitor to decay, triggering the on-chip voltage supervisor resulting in a system reset. The *RN BMOD* was not sufficiently accurate or of high fidelity to match the actual circuit behavior under such circumstances. Hence, the DMS test cases were always passing. However the first AMS co-simulation of the scenario failed. The failure itself was root-caused to be due to an inappropriate delay during the switching due to incorrect firmware handling of the reference switching scenario. This design issue or bug if not identified before tape-out would have rendered the NVM IPs completely untestable. This was resolved by an appropriate software function to perform fast LDO switching. Further the *RN BMOD* was updated to reflect proper behavior to avoid or minimize future iterations of costly and time consuming AMS co-simulation.

B. Common Random stimulus Generators

The constrained random stimulus was used for a) IO functionality checks using stream of asynchronous events and b) on-chip voltage monitors/supervisor checks using random voltage variations as shown in Figure 10.

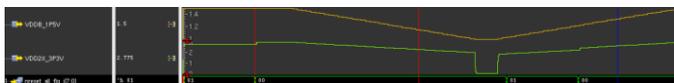


Figure 10. Random input variations

This helped to identify multiple analog issues which include erroneous voltage monitor and reset thresholds.

C. Scan IDDQ AMS Co-Sim. Made Possible

The MS SoC functional power-up sequence is as shown in Figure 11. VDD_IN is the only external supply and VDD_OUT is connected on board to VDD. VDD, VDD_DIG, VDD_MEM and VDD_ANA are all powered on-chip. A normal scan IDDQ test scenario is illustrated in Figure 12. On-chip switching regulator was enabled by default at power-up, though it is not required during IDDQ testing. All on-chip supplies were disabled and bypassed with external supplies for appropriate IDDQ levels.

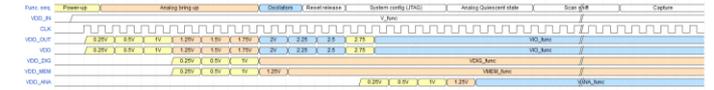


Figure 11. SoC functional power-up & scan test scenario

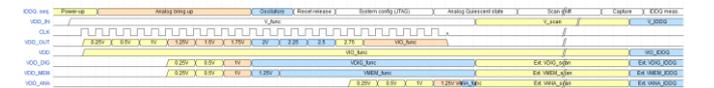


Figure 12. SoC scan IDDQ test scenario

Due to typically long scan chains in SoC such simulations take long time even with event driven simulators with digital abstraction. However, any high current in analog domains due to inappropriate configuration or defects may involve unacceptably long post silicon debug effort affecting the time to market adversely. Such scenarios are usually not simulation using AMS co-simulation due to the high cost and simulation times involved. Using appropriate optimization techniques described in [10] a slightly modified scenario to significantly reduce runtime without affecting the core focus of the test case was arrived at as shown in Figure 13. A simulation runtime of more than 8 weeks was thus reduced to about 12 hours. This helped in identifying the root-causes of high IDDQ levels and to appropriately fix the design or arrive at an optimal system configuration even before the SoC tape-out. Thus the approach helped in significantly reducing if not avoiding the post silicon test debug and optimization iterations to bare minimum.

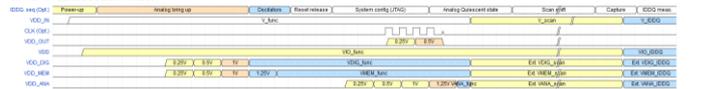


Figure 13. Optimized IDDQ test scenario

D. Advantages of detailed visibility due to DMS and AMS co-simulation

Detailed visibility into SoC and individual IP blocks helps to identify weaknesses and problems with test solution that are very difficult to debug otherwise. Real production test cases helped to detect and fix design issues in DFT logic that were not otherwise covered by base verification plan. This helped in fixing the LDO switching bug in firmware that would have otherwise rendered NVM BIST unusable and untestable on silicon; detecting obscure bug affecting corner trim cases by using random seeds to simulate PVT variation; identifying sources of excess leakage current seen in IDDQ measurement; successfully identifying and resolving a major yield issue gating entitlement cost; identifying sources of IR drop affecting test measurement accuracy. Overall it also resulted in a high level of test maturity at tape out, easier test debug and reduced cycle time, thus faster time to market. 90% of NVM tests working out of box on 1st silicon with all-new IP, DFT and SoC platform. It further helped achieve first customer samples shipped within 1 month with very high level of test coverage. On subsequent spin, high level of test maturity allowed large quantity of customer samples shipped within weeks of 1st Si with complete NVM test coverage. Overall it enabled a test strategy discussion

much before tape out to improve DPPM. Otherwise problem would have been identified much later during silicon test stage and resulted in a reactive approach to test and qualification.

IX. RESULTS

The simulation flow defined in this paper is used in the development of our mixed-signal micro-controller with RF integration shown in Figure 9 and as discussed in section VIII. Multiple design issues were identified using unified (DMS & AMS co-simulation) flow. The SoC level AMS co-simulation progress is shown in Figure 14. Some of the key outcome and execution metrics are listed below, which is also illustrated in Figure 15.

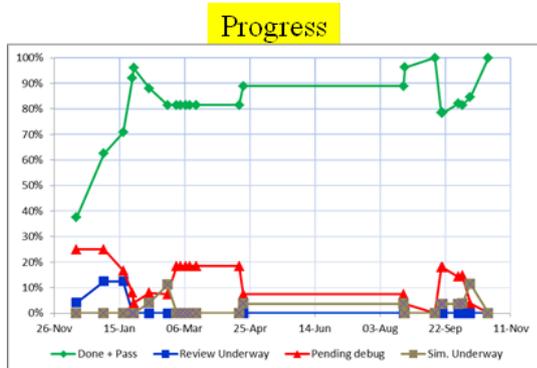


Figure 14. SoC AMS co-simulation progress

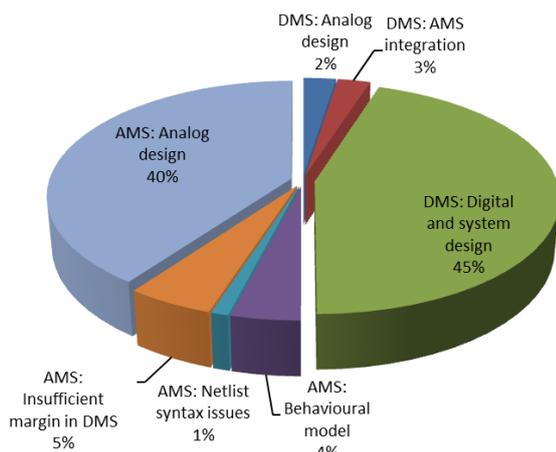


Figure 15. Pre-silicon design issues identified by DMS & AMS co-simulation

- Most design and integration issues identified at RTL stage
- 10% of DMS items are analog design or AMS integration issues
- 10% of AMS items are related to incorrect behavioural models, netlist syntax issues at early stages
 - Reduced subsequently after enhancing IP release QC mechanism
- 10% of AMS items are due to insufficient margin in analog checks for DV (DMS)
- 50% of all issues were identified by AMS
 - Key analog issues include erroneous reset thresholds, on-chip supply over loading, higher low power/shutdown mode current consumption, default trim/calibration effectiveness, analog test bus and DFT issues,

NVM firmware, DFT, trim & test sequence issues

- Resulted in single verification environment for digital, AMS and software verification
- Verification quality was maximized in a very minimal schedule
- Established close collaboration between AMS and DV team which in-turn helped both to understand the respective domains thoroughly
- DMS & AMS utility extended successfully to test readiness & silicon issue debug root-cause analysis

X. CONCLUSION

In this paper we described a unified verification methodology and environment for DMS and AMS co-simulation based verification of AMS SoC. The key feature of the framework is the ability to reuse most of the existing DMS TB infrastructure and utilities, enabling functional coverage closure for AMS co-simulations, avoiding issues at late in the design cycle using automated checks and early closure of AMS co-simulations with shorter run times. An application of the proposed methodology to an industrial case study was presented with significant improvements in quality and efficiency of pre-silicon verification.

ACKNOWLEDGEMENTS

The authors thank Maneesh Soni, Rajeev Suvama, Ramu Narthu, Ralf Hueffmann, Dirk Preiksart, Oliver Nehrig, Debasish Paul and Ibukun Olumuyiwa of Texas Instruments Inc., Zoran Koprivica, Srdjan Pjano, and Nebojsa K of ELSYS Easter Europe, and Giuseppe Scata of Apple for their contributions to the development of the methodology and execution of the same during the aforementioned product development.

REFERENCES

- [1] "System Drivers: Mixed-signal Evolution," ITRS, 2011.
- [2] Lakshmanan Balasubramanian, Puneet Sabbarwal, Rajesh Kumar Mittal, Prakash Narayanan, Ranjit Kumar Dash, Anand Devendrá Kudari, Srikanth Manian, Sudhir Polarouthu, Harikrishna Parthasarathy, Ravi C Vijayaraghavan, Sachin Turkewadikar, "Circuit and DFT techniques for robust and low cost qualification of a mixed signal SoC with integrated power management system," Design Automation & Test in Europe Conference & Exhibition (DATE) 2011, pp. 1-4, 2011..
- [3] Rajesh Mittal, Lakshmanan Balasubramanian, Adesh Sontakke, Harikrishna Parthasarthy, Prakash Narayanan, Puneet Sabbarwal, Rubin A. Parekhji, "DFT for Extremely Low Cost Test of Mixed Signal SOCs with Integrated RF and Power Management," 2011 IEEE International Test Conference, pp 1-10, 2011.
- [4] Rubin A Parekhji, Michael Nicolaidis, "SOC Design Readiness for Automotive Applications," Design Automation & Test in Europe Conference & Exhibition (DATE) 2013, pp. 1-4, 2013.
- [5] Giuseppe Scata, Ashwini Padoor, Vladimir Milosevic, "Accelerated SOC Verification Using UVM Methodology for a Mix-signal Low Power Design," Design and Verification Conference and Exhibition Europe (DV Con Europe), Oct. 2014.
- [6] Lakshmanan Balasubramanian, Ruchi Shankar, Atul Ramakant Lele, Vijay Kumar Sankaran, Sharavathi Bhat, Vinaykumar S Hegde, "Surprise or Shock? Transistor level functional analysis of digital circuits and systems are still needed!," CDNLive India, 2014.

[7] Lakshmanan Balasubramanian, Bharath Kumar Poluri, Shoeb Siddiqui, Vijay Kumar Sankaran, "Efficient methods for analog mixed signal verification: Interface handling methods, trade-offs and guidelines," Design and Verification Conference and Exhibition India (DV Con India), 2014.

[8] Lakshmanan Balasubramanian, Muruges Prashanth Subramaniam, Atul R Lele, Ranjit K Dash, "Automated correct-by-construct methodology for RTL design and analog mixed-signal test bench generation enables early design closure of mixed-signal SoC," Design and Verification Conference and Exhibition India (DV Con India), 2014.

[9] Lakshmanan Balasubramanian, Bharath Kumar Poluri, Venkatseema Das, Vijay Kumar Sankaran, Aman Gupta, Badrinarayanan Zanwar, "Surviving eternal conflict of accuracy and cost: Efficient Analog Mixed Signal Co-simulation with Spectre-XPS-MS," CDNLive India, 2015.

[10] Vijay Kumar Sankaran, Lakshmanan Balasubramanian, Nadeem Husain Tehsildar, Bharath Kumar Poluri, Tutorial on "Recent trends in AMS co-simulation for low power focus and efficiency improvement," Design and Verification Conference and Exhibition India (DV Con India), 2017.

[11] Lakshmanan Balasubramanian, Bharath Kumar Poluri, Aman Gupta, Vijay Kumar Sankaran, Badrinarayanan Zanwar, "Circuit check in SoC AMS for a robust, reliable low power implementation," CDNLive India, 2015.