Design and Verification Challenges with Third Party IP Cores

Ruchi Shankar
Connected MCU
Texas Instruments (India) Pvt. Ltd.
Bengaluru, 560 093, India
r-shankar@ti.com

Prachi Mishra
Connected MCU
Texas Instruments (India) Pvt. Ltd.
Bengaluru, 560 093, India
p-mishra@ti.com

Abhinav Parashar
Connected MCU
Texas Instruments (India) Pvt. Ltd.
Bengaluru, 560 093, India
abhinav.p@ti.com

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

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

Abstract— **IP reuse is all about improving productivity and can result in significantly shrinking the design cycle time especially with configurable third party IP cores. Increasing amount of third party IPs find their way onto today's complex system-on-chip (SoC) designs. Hence it is paramount that designers build a large and expanding knowledge base incorporating lessons learned out of accumulated experience from several of designs containing a broad range of IP blocks into tangible design, verification and test methodology components. These components include checklists, automated IC analysis programs, and processes both internal and collaborative. This knowledge base is usually combined with the experience of the individual IP and EDA vendors to ensure the lowest possible risk to each design. Integrating third party IP core typically involves various challenges. These challenges involve compatibility with power, reset and clock (PRC) schemes, design methods used to achieve system low power goals, integration scalability, and design verification methods to achieve comprehensive entitled coverage. Resolving them requires additional design, integration and verification effort. Design verification (DV) in general could be more challenging, as most third party IPs are verified in isolation agnostic to the context of the system. Ensuring that the third party IP cores as used in the SoC will ultimately meet all requirements is a highly complex task that requires a dedicated, expert team with an explicit focus and responsibility towards this task. This paper outlines design and DV challenges and resolution in integrating third party IPs in today's high-end ASICs/SoCs.**

Keywords— *Third party IP, MCU, IOT, EDA tool, SoC, CDC, RDC, DV*

I. INTRODUCTION

At a semiconductor design house, it perhaps makes the most sense to focus only on the development of IPs in-house which differentiates a product rather than doing whole thing on its own. For a low power SoC which is directed towards portable, battery operated, autonomous embedded internet-of-things (IOT) market with related application requiring low power and low cost as the DNA for all underlying building blocks[1][2][3][4], it is utmost important to do the analog, RF, and power management (PM) blocks in-house as those will critically differentiate the product from their competitors in the market. This motivates the need of procuring IP Cores from third party IP vendors. IP procuring is reflected as an effective choice with various benefits specifically, better time to market, lower cost, increased focus and reduced risk. Nevertheless, this approach imposes difficulties due to specification compromises, limitation with modification

rights which can possibly impair a company's flexibility and agility to address dynamically evolving requirements, if not taken into account well in advance. Integrating third party IPs is rapidly becoming one of the biggest challenges in the SoC/ASIC industry. This mandates convergence and integration of third party IP cores along with analog mixed-signal (AMS) contents, and power management. This situation brings complex challenges in design and verification[5] that are associated with usage of third party IP cores.

The rest of the paper is organised into 5 sections. Section II describes the design challenges complexity with third party IP cores and their respective resolutions. Section III details verification challenges complexity in using third party IP cores and describes how these challenges are overcome. Further discussion on wider application and future scope is dealt with in section IV. Section V concludes the paper.

II. DESIGN CHALLENGES & PROPOSED SOLUTIONS

Using third party IPs poses different ranges of challenges encountered at IP design and SoC integration phases. These challenges include the incompatibility of third party IP cores with a) system level power, reset and clock (PRC) management scheme; b) most commonly used coding practices as per the EDA vendor expectations on LINT checkers; c) the essential pseudo static assumptions (along with enable conditions); d) system level low power design requirements and associated functional cover points for proving the convergence scenarios; e) scalability and configurability needs including for memory mapped registers. This requires additional design effort to a) add wrapper level logic and functionality, b) abstract out sequences with PRC manager and c) create protocol converter bridges to map the legacy protocol definition of third party IPs to system level protocol definition. It is essential to address the comprehensive closure of pre-release quality checks (QC), EDA tool vendor support and related dependencies. We describe each of these challenges along with the respective proposals to resolve them in subsequent portions of this section.

A. Power

Typically third party IPs, CPU or non-CPU IP, have internal reset synchronisers. Power un-gating usually involves a strict sequence of events comprising reset release, followed by retention release, followed by power gating

isolation release. Typically, the clocks remain gated off to the CPU for this entire sequence as showed in Figure 1 below. The reset synchronisers within third party CPU IP enforces the clock manager to have clocks running during the power un-gating phase of the sequence. This is needed to enable retention in third party IP by ensuring a proper reset release handshake is achieved prior to de-asserting retention control. In addition to affecting the handshaking mechanism, this requirement also impacts and dictates the functional and electrical parametric specification of the library retention cell(s).
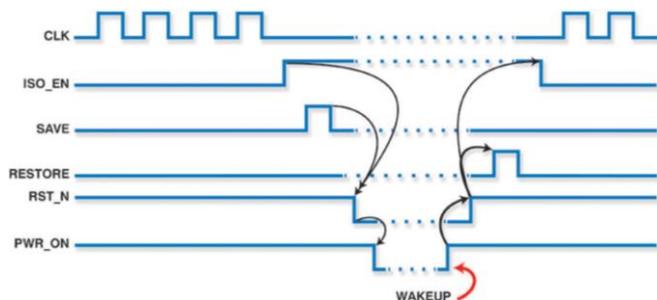

Figure 1 Power gating & power un-gating

RESOLUTION

A custom reset handshaking sequence prior to the release of retention control by PRC controller is implemented to overcome this challenge. Additionally this necessitates un-gating the clock for brief duration after power un-gating and reset de-assertion as shown in Figure 2 below. Targeted library retention cell is developed to ensure that the presence of free running clock during retention state will not lead to the loss of retained state.
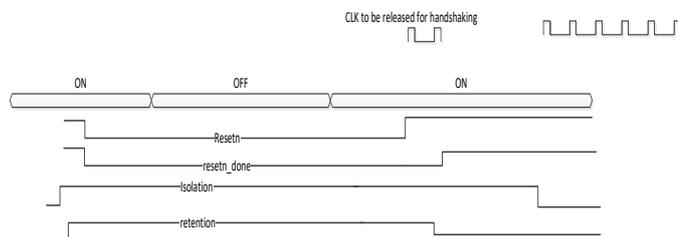

Figure 2 Reset handshaking

B. *Reset*

Some non-CPU third party entities implement synchronous reset. We will consider an SoC design implementing asynchronous reset assertion while synchronous reset de-assertion. Sequence applied for disabling an IP typically have the reset isolation asserted followed by reset assertion. In a design with asynchronous reset implementation, reset assertion is maintained for couple of clock cycles by the reset manager. In third party IP, reset input goes through data synchroniser. So, for the reset to propagate through the synchronising flops, clock is needed and reset input is required to be asserted for N clock cycles depending upon the density of data synchronisers usage even post asynchronous reset assertion from the reset manager. This imposes special hardcoded reset sequences to be designed for such IPs as against a generic reset controller definition.

RESOLUTION

To overcome this challenge, an appropriate reset handshaking with power manager is implemented in addition to a clock management scheme to keep the clocks available to the third party IP until reset done indication is asserted. This can be achieved by keeping a counter within the power manager to keep the clocks active for N clock cycles which will be derived from the depth of the data synchronisers within third party IP.

C. *Timing Closure:*

Many target applications have very aggressive specification requirements for low power and high performance to achieve best in class devices. This gets translated to meeting frequency targets at different operating point conditions. The recommended input and output delays for a third party CPU IP are close to ~40% of the clock cycle which leaves out very little room for the wrapper logic necessitated by integration context. Consider implementing instruction and data cache around third party CPU IP. Closing timing paths involving clocked memories is quite challenging and tends to impact either the frequency of operation or system performance by making such paths multicycle. Additionally, if the platform has to meet safety standards then deploying single error correction and double error detection (SECDED) mechanism makes the timing closure almost impossible unless variation in error detection techniques are employed.

RESOLUTION

For safety critical designs, where ECC implementation is required, one can segregate error detection and error correction in different clock cycles to reduce the cone of logic. Clock skewing techniques in SoC can be employed to ease timing closure. The worst case solution would be to reduce the frequency of operation or making such timing critical paths multicycle, both of which will degrade the system performance.

D. *Memory Mapped Register Definitions*

Third party IPs are characteristically highly configurable as they need to support variety of feature sets. They typically implement the memory mapped registers as R/W (Read/Writable) which are not desirable for certain configurations. This implementation imposes significant challenges on the IP verification to enable negative testing. Negative testing is needed to verify and ensure that these memory mapped registers, when written by software, will not lead to any unintended functionality.

RESOLUTION

This challenge is overcome by implementing redundant address decoding logic as a wrapper around third party IP to ignore write operations and ensure reading zero for such memory mapped register bit definitions.

E. *EDA Vendor Support*

Configurability of third party IPs poses challenge in closing the basic quality checks like LINT, and comprehensive verification targeting clock domain crossing (CDC), reset domain crossing (RDC), and low power aspects. The signoff for these quality checks vary across different EDA vendor tools. It imposes difficulties due to

incompatibility in formats of constraints and waivers, thus towards adapting them between different EDA vendor tools and flows.

On top of this, third party IPs typically implement the reset tree and clock tree deep down the hierarchy which are usually not allowed to be modified. Moreover, they don't recommend taking care of the reset domain crossings in hardware as shown in Figure 3 below. Such RDC violations needed to be identified and resolved comprehensively at the SoC level. Some of the most critical quality checks are related to CDC, RDC and low power.
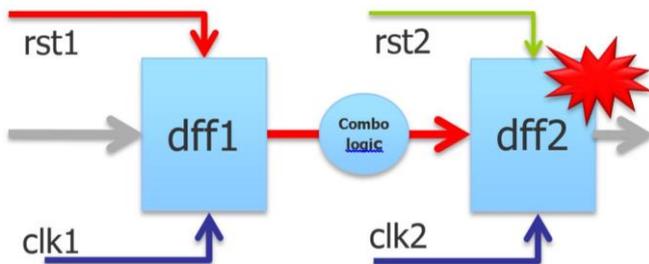

Figure 3 Reset domain crossing

Mismatch between the coding styles and tools expectations results in challenges with RTL lint quality checks. One of the common issues in RTL lint is the width mismatch between source (RHS) and target (LHS) arguments of logical operations. Two such instances are illustrated in Figure 4 and Figure 5 involving scenarios with LHS > RHS and LHS < RHS. In this instance example they are false violations and hence have been waived.

wire [16:0] mul16_fixup_cx = {({17{mul16_in1_cx[15]}} & ~{2'd3,mul16_in2_cx[14:0]}) + ({17{mul16_in2_cx[15]}} & ~{2'd0,mul16_in1_cx[14:0]}) + mul16_in1_cx[15] + mul16_in2_cx[15]};

Figure 4 LHS and RHS mismatch causing false violation

wire unused = (|dpu_trc_cx_dphase_i[3:2]) | dpu_trc_base_q_i[2] | d_htrans_i[0] | d_hsize_i[2];

Figure 5 Unused wire declarations in third party IPs

RESOLUTION

A functional check to find an occurrence of overflow is employed as a preferred solution to comprehensively analyse such issues than using a structural lint check. This would allow the violation to be reported only in case there is functional failure resulting in an overflow of a valid data causing unintended functional behaviour. Additionally critical and comprehensive set of constraints, assumptions and waivers are compiled and aligned between IP vendors and system integration/design houses to correctly disposition these violations. Appropriate assertions are employed to identify them automatically during dynamic simulations.

Input clock to the IP is suppressed for the period when the lower order reset gets asserted to avoid activity on destination flops that are on controlled by higher order resets. To enable this, PRC manager provides reset isolation control so as to get asserted at least 1 clock cycle in advance as compared to actual lower order reset assertion. Furthermore, verification scenarios are added to ensure that the higher reset domain functionality is not impacted.

F. Bridge Development for Protocol Conversion

Protocol support from third party IP is typically limited to legacy bus (AHB) protocol usage. SoC normally implements standard or proprietary protocols that offer flexibility in selectively adding the pipelining at the entry or exit points in a matrix (cross bar with multiple master and slaves) to ease timing closure.

III. VERIFICATION CHALLENGES & PROPOSED SOLUTIONS

Verification of a system with third party IP integration presents several diverse challenges. Different issues with their solutions are discussed herein.

A. Power Sequence Verification

A third party IP core in a subsystem typically have the reset tree implemented deep down the hierarchy that is not allowed to be modified. Power sequence verification is challenging in cases of a third party IP core with internal reset synchronisers in the reset tree. To ensure that during power up these synchronisers take proper value before retention is removed, clock to these synchronisers has to be active while retention is still asserted and they should indicate a reset done handshake before power sequence moves ahead to de-assert retention. Verification has to ensure that the sequencing does not cause any issue at either the IP or the subsystem level.

RESOLUTION

A subsystem level verification test-bench is used to verify the IP in the context of higher level of integration. To ensure that all possible scenarios are covered, an IP level verification setup involving appropriate bus function models (BFM) is used. In this approach, the IP core is abstracted to a BFM with only power management interface. This BFM is then connected to the power management unit and fully randomised, with appropriate cover points. The BFM is coded to address the power sequencing expectations, including retention and reset handshake. Advanced retention behaviour is modelled in the core simulator to ensure appropriate clock and reset dependency on the retention behaviour.

B. Parameter Verification

A third party IP can have configurable parameters in the RTL which can be used to optimise internal modules in terms of what functionalities they implement. These third parties IPs will have a wrapper around them which will be integrated at SoC level. Here, the SoC may have expectations on what parameters can be reconfigured from top level, as well as those which are not controllable and are hardcoded inside the wrapper. Verification has to ensure the correctness of parameters.

RESOLUTION

The change of parameters inside the third party IP is usually reflected in registers readable from the IP, as well as through a relevant change in functional behaviour. Therefore, a dedicated test suite is developed with parameterised test-bench. This test suite checks for points of interest in terms of minimum, maximum and typical values of the parameters and thus ensures sufficient re-configurability at SoC level.

C. RDC Analysis Assumptions

A third party IP with multiple clock domains typically has reset synchronisers in reset tree implementation. Additionally, power domains existence within the IP leads to multiple derived resets and clocks. During RDC analysis, assumptions are made by the designer on categorising all these resets in a specific order in the CDC constraints file, i.e. from the highest order reset to the lowest order reset. These assumptions are required to be verified and proven as there could be errors due to lack of detailed visibility into or sufficient knowledge about the third party IP.

RESOLUTION

The reset order assertions are proven in a formal verification setup to ensure that there are no circumstances under which the expectation on the reset ordering is violated. For example, in Figure **6**, *reset1* is the highest order reset and *reset2* and *reset3* are lower order resets derived from *reset1*.

```
assert property (@(posedge test_clock) ##2 ~(reset1) |-> ~(reset2));
assert property (@(posedge test_clock) ##2 ~(reset1) |-> ~(reset3));
```
Figure 6 SV assertions checking the reset ordering of the IPs
Fig 6:

Figure **7** shows cover points for both assertion and de-assertion of resets.

```
cover property (@(posedge test_clock) ##2 ((reset1) & $rose(reset2)));
cover property (@(posedge test_clock) ##2 ((reset1) & $rose(reset3)));
```
Figure 7 SV cover points for assertion and de-assertion of resets
Fig 7:

D. Proving RDC Crossing Design Fix for Third Party IP

Typically third party IP does not recommend taking care of reset domain crossing within the IP. The required logic is implemented in a wrapper around the IP to enable suppression of the input clock(s) to the IP. This is usually done using a pre-qualifier condition like reset isolation. This imposes huge challenge on DV in ensuring that the clock suppression to higher order reset cone of logic works properly without any impact to intended functionality.

RESOLUTION

In DV, clock gating checks are added to prove that there are no clocks available on flops associated to higher order domain resets whenever a lower order reset gets asserted. It is of utmost importance for comprehensive verification to identify all feature sets within the IP that belongs to higher order reset(s). Functional cover points are mandatory for each of these feature sets exercising clock suppression scenarios during lower order reset assertion. It becomes more challenging when the SoC implementation has supplementary logic around the IP that may belong to same higher reset domain as of the IP. In such scenarios, there will be a proportional blow up of the amount of functional cover points.

E. Convergence Issues

There are scenarios within third party IPs where many (N) independently synchronised signals are converging into a common decoding logic. Figure **8** shows generic view of convergence issue. This may impact functionality depending upon when the data is getting stabilised for each of the arcs in presence of meta-stability (which may be 2 to 3 cycles).

Because of this uncertainty, unreliable decoded data may get sampled and propagated through the design.
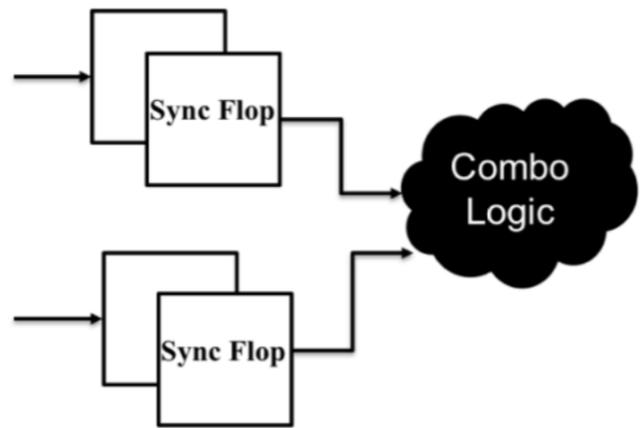

Figure 8 Convergence issue inside the IP

RESOLUTION

Functional coverage is typically done as part of IP verification. However with convergence arcs, the number of scenarios can explode. Hence there will be a need for dedicated scenarios for functional coverage or increasing the different number of seeds for random verification scenarios. To ensure the convergence issue is taken care, detailed coverage analysis is needed, exercising the existing test suite with meta-stability injection.

F. Verification of System Level Bug Fix

Sometimes the architecture level limitations impose such constraints to handle any design bug fixes by updating the IP configurations or by doing the internal design changes. Since the knowledge base for the third party IP is very limited, the completeness of bug fix verification adds lots of additional challenge compared to any in-house design updates.

RESOLUTION

Even though all IP vendors verify their IP, it is still very important for the users to re-run the vendors verification vectors to make sure there are no missing deliverables. Most importantly, there is a need for a thorough verification of the IP including the integration logic around the IP. This will help us to re-verify the IP functionality in addition to easily ensuring that there are no side effects of bug fixes since the whole verification suite is available in-house and verification engineers are also experienced in handling the third party IP regressions.

G. Lack of IP Ownership

IP licenses come with various restrictions which can "get in the way". Examples are: reuse, disclosure and modification rights limitations which can possibly impair a company's flexibility if not taken into account well in advance.

RESOLUTION

This needs to be thought well in advance and engineers should be trained up front with in-depth knowledge.

H. RTL and GLS Mismatch

A third party IP that works perfectly in RTL, can still have timing and functional bugs hidden behind macro definitions (*ifdef/pragma*) which can be sensitised only in synthesis and/or gate level simulations (GLS). These quiet IP bugs can be catastrophic.

RESOLUTION

The GLS regression should cover major functionalities associated with third party IPs which can catch these kinds of issues.

I. Comprehensive Verification of Electrical Specifications

Electrical specifications sign-off through appropriate SPICE level simulations across process, voltage and temperature (PVT) corners and all boundary conditions is critical especially for analog and mixed-signal IPs. However, it can be counter-productive to completely rely on an IP level sign-off for reasonably complex AMS IPs and sub-systems due to the possibility of incomplete specifications, sub-system level simulation sign-off and unstated assumptions on the sub-system level integration especially when sufficient visibility into the detailed specifications of the components of the sub-system and IP/sub-system level verification. Any gaps thereof can result in silicon issues consuming exorbitant post-silicon debug effort; avoidable silicon re-spins related costs and delays.

RESOLUTION

To mitigate such risks, SoC integration team gaining a sufficiently detailed understanding of the architecture of the AMS sub-system, its components, and integration assumptions are critical. This can be gained through detailed design and integration specification documents, verification plan including SPICE level simulation details and test conditions as a part of IP delivery mechanism. In the absence of the same, it is necessary to treat such AMS IPs as having gone through insufficient verification and plan for the same at SoC level to ensure it is verified in the right context and under valid conditions.

CASE STUDY

One of the examples involves a third party AMS sub-system with an on-chip LDO powering several analog/RF modules. In the absence of any detailed verification plan shared by the third party IP vendor, the authors executed a critical set of AMS co-simulation based verification at the SoC level with the third party sub-system completely in SPICE configuration. Though initial simulations took exorbitantly long simulation runtime, it helped identify a design integration weakness causing the LDO not to power-up correctly in one PVT corner, due to overloading beyond its current capacity by the analog sub-system. A detailed debug resulted in identifying sections of the analog sub-system remaining enabled that are not necessary to be power-up at that point in the power0-up sequence. This triggered a design change to appropriately sequence the enabling of related sections of the analog sub-system. In the absence of any other issue found through simulations, the shorter simulation test-cases to verify only that known design weakness are kept for subsequent regressions.

J. Mismatch in Power Intent Format & Low Power Verification

Power intent (PI) is a design artefact used for specifying low power design requirements. There are two standard formats compact power format (CPF) and uniform power format (UPF) being used for the purpose. However, there have been critical differences among these standard formats, the consistency of interpretation and support by various EDA tools across domains and EDA vendors [6]. Inconsistency in choice of PI format at IP and SoC level can cause not only practical execution difficulties, but also quality gaps due to the aforementioned limitations with EDA tools. Tactical solutions used to address the known tool limitations and inconsistencies can pose additional challenges.

RESOLUTION

In the absence of clear EDA support models for different PI format to be co-exist in single design environment, it is advisable to choose one and ensure all required IP vendors have appropriate plan, including bridging any competency gaps, to support the same in a timely manner. In addition to the choice of PI format, it is also required to come up with a clear SoC level power integration, verification and implementation strategy, compile and align on detailed set of guidelines and recommendations on the coding style, required details to be captured in the power intent, acknowledging the known tool inconsistencies and tactical solutions to address the same. Though there is no escape from the use of a small set of tactical solutions for known gaps, it is highly recommended to keep it a small set, understand their effects on all domains, and have alternate QC mechanisms to address any consequent quality gaps.

IV. DISCUSSION & FUTURE DIRECTIONS

Persistent feedback needs to be provided to third party IP cores wherever an improvement can be made possible in the source code.

Few examples could be improving coding practices as per the lint violations, taking care of reset domain crossing internal to third party IP by enabling necessary hooks at the entity like reset isolation port or clock gate enable, and providing an option to remove reset synchronisers or data synchronisers (in case the reset de-assertion is synchronous to input clock of third party IP core) at SoC integration phase.

For the reserved bits, third party IP should implement masking so as to ensure that write is ignored and read returns zero. Automated test suite/VIP controlled by the same masking parameters should be provided to check functional correctness.

In terms of EDA tools, there can be parameters provided incorporating different behavioural versions. For example, retention based third party IP may have different expectation with respect to various signals coming to its power management interface for correct retention behaviour. If the tool provides parameters to seamlessly switch between different models with inbuilt checkers, this will hugely reduce the verification effort in a system with different types of third party IPs.

V. Conclusions

The increasing size and complexity of modern silicon systems results in a growing need for reusable and pre-verified third party IPs, such as embedded memories, processor cores, high-speed interfaces and analog IPs. Incorporating these components into a single chip can be a challenge due to the variety of different IPs and the increasingly difficult design rules for modern processes. This paper discusses some of the best design practices and methodologies that help ensure the successful integration of third party IPs into next generation, complex SoC designs, enabling an accelerated path to first pass silicon success.

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] Ashwini Kumary Padoor, Lakshmanan Balasubramanian, Ibukun O Olumuyiwa, "Unified Verification Methodology for Digital and Analog Mixed-Signal Co-Simulation for Low Power Embedded SoC," IEEE WINTECHCON, 2018.

[6] Lakshmanan Balasubramanian, Venkatraman Ramakrishnan, Zhong Fan, Vijay Kumar Sankaran, Aswani Kumar Golla, Arumugan A, Rewin Edwin, Manash Ranjan Raiguru, Tutorial on "Modern Recipes for Brewing the Inevitable Methodology for Today's ICs: Low Power Mixed-Signal Design Verification," Design Automation Conference, 2019.