A Configurable, Re-usable UVM Environment Coupled with Advanced Spice Simulator for Analog and Mixed-Signal Verification of a Display PMIC

Abstract- In this paper, we will discuss about UVM-based Analog and Mixed-Signal (AMS) Verification performed on a Power Management IC (PMIC) that was designed to power OLED mobile display panels. Earlier the focus of the verification was to perform Digital Mixed-Signal (DMS) Verification using SystemVerilog Real Number Models (SV-RNM) to cover various scenarios of the PMIC, such as basic power up and power down sequence, I2C write and read transactions, Dynamic Voltage Scaling (DVS), and testing the PMIC's protection logic against various fault scenarios. In this paper, we present how we performed AMS verification of a subset of those DMS items by re-using the UVM-based verification environment from DMS to AMS. We illustrate some key test scenarios that were verified accurately by running AMS simulations, present different mixed-signal configurations created for different scenarios, and describe how we could unearth critical design issues with the verification methodology used. We have also mentioned whether we could correlate the simulated data from UVM-AMS verification with the actual silicon data to achieve confidence on the verification performed.

Keywords: PMIC, UVM, DMS, AMS, Spice, SV-RNM

I. INTRODUCTION

Power management has become inevitable in today's System on Chips (SoCs) with low-power high-performance chips being today's norm across applications such as storage, memory, interface, display, battery management, etc. Owing to this, the need for design, verification and adoption of Power Management Integrated Circuit (PMIC) need not be overemphasized, since a PMIC has become mandatory in almost all analog-centric ICs today. The PMIC discussed in this paper is targeted for display applications in order to power OLED mobile display panels [3], camcorders, digital cameras and multimedia players. It comprises of highly efficient switching converters and a low-dropout linear regulator (LDO). Apart from these key blocks, the PMIC consists of the following blocks: (a) A high frequency oscillator (HFO) to generate clocks, mainly for triggering the digital logic and (b) Housekeeping circuits such as reference voltage (VREF) and bias current (IBIAS) generators, IO cells, ESD protection cells, pad rings, etc.

There is the core digital block that drives the enables for the converters and other analog blocks. All converters are input to output isolated, have integrated rectifiers, come with TR Segment Control (TSC) and are designed to operate from a single-cell Li-Ion battery and charger. The register banks inside the digital block (and so the converter outputs) are programmable by I2C serial communication protocol. Unlike other chips that traditionally come with single slave address for registers in the bank, this PMIC comes with two slave addresses, each with its own set of register banks. While one bank of registers are programmable by I2C protocol using the serial clock and data lines, the other bank is programmable by a separate set of lines. A GPIO input decides which set of lines will be used for I2C read and write.

The block diagram architecture of the PMIC is shown in the next page.

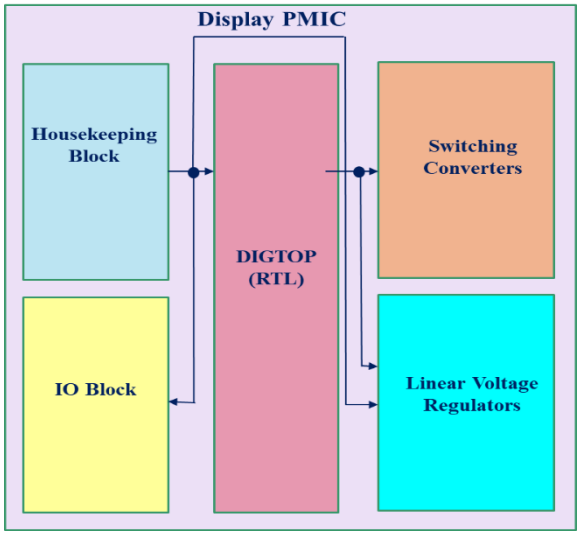

Figure 1. Display PMIC Block Diagram Architecture

With such medium to high-sensitive analog blocks in place, the normal power-up and power-down sequence of the PMIC would follow a particular order with soft-start times and delay times as indicated in the below timing diagram. Converter 1 powers up first, followed by auto-enabling of Converter 2 (with a pre-converter, viz., Converter 3 and an LDO), which is then followed by enabling Converters 4 and 5. A symmetric sequence is followed during power down. However, Converters 4 and 5 can be made to enable and start ramping up without enabling Converter 1 at all. Usually under normal start-up sequence controlled by the Display Driver Interface (DDI), Converter 1 always powers up before all the other converters.

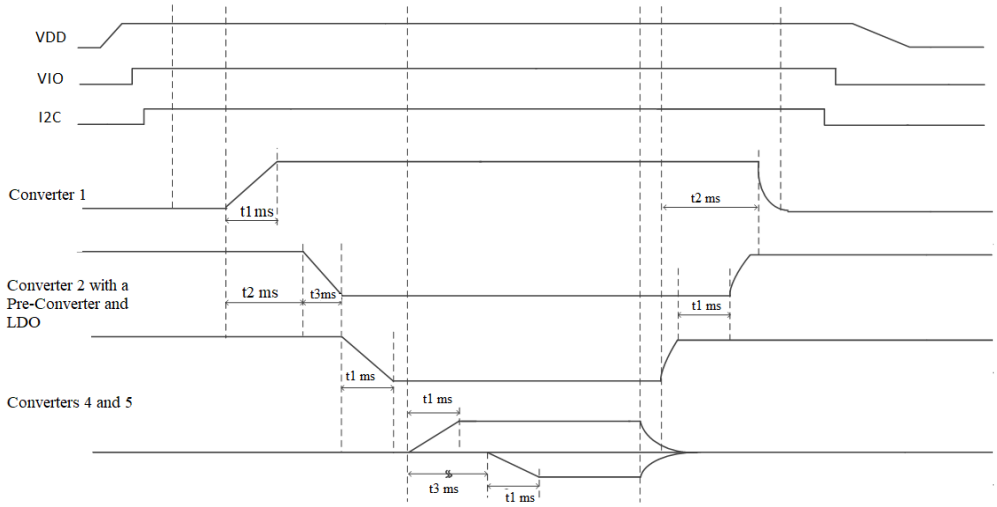

Figure 2. Display PMIC Power On/Off Sequence

The initial part of the verification of this PMIC viz., the Digital Mixed-Signal (DMS) verification, was performed by following the flow discussed in [6] and as shown in the below flow diagram. From the schematic of the PMIC chip-top level, we created a mixed-signal configuration, where we bound the blocks of interest to use analog behavioral models (ABMOD) using SystemVerilog real-number modelling (SV-RNM) for the time-consuming analog blocks [4][5]. We then extracted the netlist of this configuration, and along with RTL codes for the digital blocks and with the developed SV-RNM models for analog blocks, we compiled and simulated the design using purely event-driven logic simulator. We took care of the Real-to-Logic (R2L) and Logic-to-Real (L2R) conversions using appropriate connect modules from the simulator [12]. The models developed for analog blocks were also validated against their respective schematics by using common testbenches before using them in DMS verification [7][9].

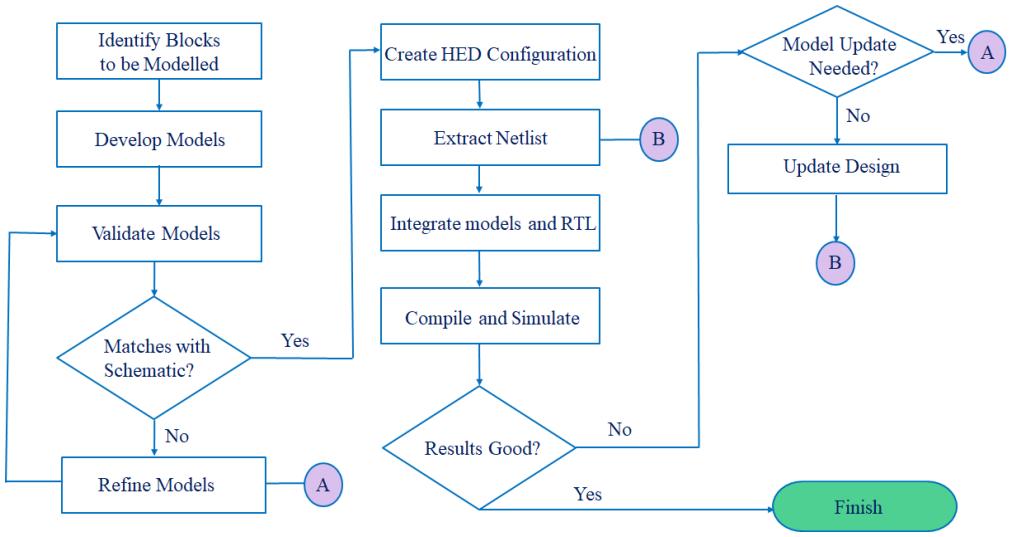

Figure 3. DMS Verification Flow for the PMIC

II. UVM-Based Verification Environment For The PMIC

Once we completed the DMS verification of all the items in our verification plan, we identified a subset of those items to be verified in AMS, by re-using the UVM-based verification environment that was used for DMS verification. A high-level representation of our UVM-based verification environment is shown below. We see that the environment encloses three key components, viz., DUT, driver and the scoreboard. The DUT that consists of both analog and digital portions of the design was verified with all analog blocks as SV-RNM models [8] during DMS verification and selected/all analog blocks as Spice for AMS verification. Initially, there were scenarios where some of the analog blocks in the DUT were selectively identified and used as Spice, while retaining the other blocks as SV-RNM models as is. This gave rise to the introduction of co-simulation since both logic and transistor-level (TL) simulators came into picture, and connect modules to take care of the conversion between signals of different datatypes, viz., Electrical-Logic-Real (E-L-R) were inserted by the simulator.

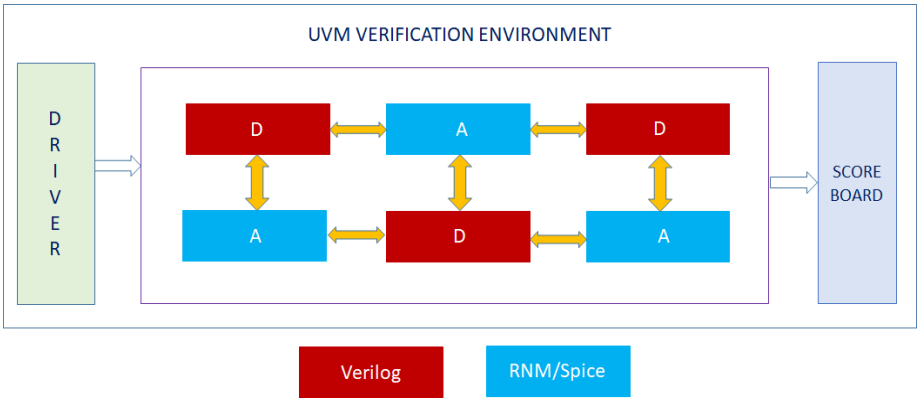

Figure 4. UVM Verification Environment

A high-level directory structure of the UVM verification environment that we used is shown below. The *compile_lib* directory holds the compilation snapshot of all the SV-RNM models and the RTL codes of the digital blocks. The *lib* directory contains SV/PSL assertions that we can add to report pass/fail status during verification, and some of the key UVM files such as base sequencer, base packages, base test, etc., that are standard for any PMIC verification and commonly used for all test scenarios. These files usually contain the basic sequences such as powering up of the DUT, enabling reset release, etc. The *models* directory is where we place all the SV-RNM models that we create for the analog blocks we identified for modelling. We then have the most important *tb* directory with two sub-

directories, viz., *top* and *env*. The *top*, as the name indicates, mainly contains the top-level testbench file that instantiates the DUT, along with instantiation of the I2C model (communication protocol) used for registers reading and writing. It also has a file for providing a virtual interface between the communication protocol and the design objects (such as pins) while accessing those objects in the UVM test cases. The *env* has all the UVM environment files such as model files that define the communication protocol, scoreboard for the PMIC, interfaces definition and instantiation, and a file that can contains some of the commonly used tasks in each test case. The *tests* directory contains the UVM test case files that we write to verify the test scenarios as per the verification plan document. The *verilog_dut* contains the RTL files for the digital blocks.

```
|-- compile_lib
|   |-- blk_main_model_libs
|   |-- blk_main_rtl_libs
|-- connectivity
|-- lib
|   |-- assertions
|   -- common
|-- mconf
|-- misc
|-- models
|-- tb
|   |-- env
|   |-- top
|-- tests
|   -- main_tests
|-- verilog_dut
-- vm
    |-- misc
    |-- scripts
    |-- vplan
    -- vsif
```

Figure 5. Directory Structure of the UVM Verification Environment

In the below diagram, we zoom into the Display PMIC chip top-level (our DUT) that shows some of the key aspects that become part of the verification activity. We performed functionality check of the analog blocks as the foremost requirement, along with which we covered the D2A & A2D connectivity checks. These were static checks that were done to make sure the pin connections between the digital top block and different analog top blocks were as expected. The test scenario to be verified was passed as UVM test case via the serial communication protocol. Using the UVM test case, we would write the required serial clock and data lines (or separate set of lines as decided by GPIO input) to program the registers in the digital, thereby enabling the required design configuration for verification.

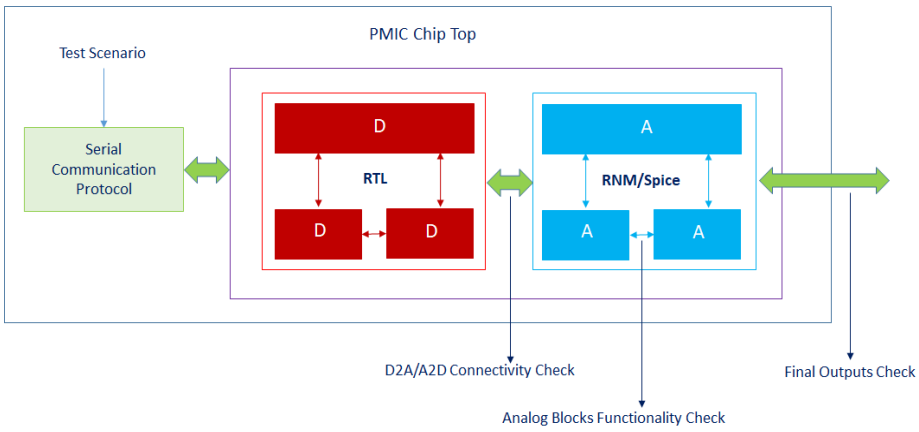

Figure 6. DUT and Communication Protocol with Verification Aspects

Depending on the test scenario to be verified in AMS, we replaced some or all of the analog blocks from their SV-RNM models with their equivalent Spice/TL abstraction as explained in Table I in the following section of the paper. This mixed-signal configuration gave rise to the need of adopting an advanced Spice simulator to simulate the Spice portion. We also leveraged the multi-threading (or multi-cpu) capability available in the Spice simulator to achieve performance (measured as runtime) [10]. We thus ran AMS co-simulation with insertion of appropriate connect modules for A2D and D2A conversions [11] between Spice and digital blocks.

Some of the key benefits that we achieved by re-using the UVM verification environment from DMS to AMS are listed below:

- Replacing the key analog blocks from their respective SV-RNM to equivalent Spice representation 'in-situ', thereby having to avoid creation of new environment for AMS verification. This was imperative because creation of a new verification environment indeed would become a time-consuming task. Environment re-use enabled starting of AMS verification on time, without having to spend any additional cycles given the tight project schedule.

- Re-use of UVM assertions and checkers that were written for DMS simulations, since the AMS verification items were a subset of DMS items. All checkers related to voltage checks, timing checks, and especially status register checks (which are huge in number) were retained and used as is.

Replacing the blocks from their SV-RNM to Spice representation and running AMS simulation comes at the cost of increased simulation runtime (performance). However, such an impact on simulation performance is outweighed by the immense benefits of getting golden results with silicon accuracy by running AMS simulation.

III. Analog and Mixed-Signal (AMS) Verification Of The PMIC

Upon completion of DMS verification of the above PMIC for various functional scenarios, such as I2C read/write, power up/down sequence, protection/interrupts, dynamic voltage scaling (DVS) and connectivity checks, requirement came up to perform AMS verification by simulating some of the key analog blocks in their Spice representation. The items that we identified for verification in AMS are listed below.

Power up and power down sequence: The first and foremost scenario to be validated in AMS for any PMIC would be the default power-on and power-off sequence. We saw the basic power sequence to be achieved by our PMIC in figure 2. Verification of this sequence not only includes verifying the switching converters' voltage regulation values, but also includes checking whether the timing specifications are met. Since we are re-using the DMS verification environment for AMS, we needed to incorporate certain changes in the stimulus. One such essential change is incorporating ramp on the supply. For DMS verification, since we used logic simulator, it was ok to make the supply go from low to high in the logical sense without involving a ramp. However, in AMS simulation involving Spice simulator, such sudden jump on the supply lines would lead to convergence issues in the simulation. Therefore, the very first update in the re-used environment was to slowly ramp up the main supply from zero to its final value (and so the testcase duration would slightly increase owing to such ramp). However this too had to be done with a careful balance of runtime vs convergence trade-off because if the ramp is too steep (or too slow ramp), the simulation would take huge amount of time for the power-up to happen. This would in turn affect the overall verification turn-around time. By using optimum Spice simulator options and multi-threading feature, we were able to also get the right accuracy vs performance trade-off while running AMS simulation.

Coming to the mixed-signal configuration for this scenario, we started by first replacing the housekeeping blocks (comprising of VREF and IBIAS generators) from their SV-RNM representations to their respective Spice abstractions. This became imperative because since we planned to keep some or all of the switching converters in their Spice abstractions (depending on the test case as shown below), the required VREF and IBIAS values to the converters coming from the housekeeping circuits had to be accurate when the switching converters are kept in Spice. If the housekeeping blocks were kept as SV-RNM models, then conversion elements (or connect modules) would be inserted between the models and Spice, which can lead to inaccurate current flowing through the path. This would in-turn affect the efficiency of the switching converters thereby leading to drop in the regulated voltage values. The below table shows the mixed-signal configurations for various test cases that fall under this verification scenario.

With the switching converter(s) in Spice, it also became necessary to add the required inductive and capacitive loads for the converters as per the application circuit provided in the PMIC specification document.

TABLE I
MIXED-SIGNAL CONFIGURATIONS FOR VARIOUS POWER ON/OFF TEST CASES

S.no	Test Case	Blocks in Spice	Blocks in SV-RNM
1	Power-on/off sequence with selection of serial clock and data lines by GPIO	- Switching converter 1 - Housekeeping blocks	- Switching converters 2 to 5 - IO blocks - Miscellaneous blocks
2	Altered power-on/off sequence with selection of separate set of lines by GPIO	- Switching converters 1 to 5 - Housekeeping blocks - IO blocks	- Miscellaneous blocks
3	Power-off using main supply and IO block supply ramp down	- Switching converter 1 - Housekeeping blocks - IO blocks	- Switching converters 2 to 5 - Miscellaneous blocks

Register write and read operation: Once we validated the basic operation of the PMIC in AMS by ensuring the default power on-off sequences, the next task was to check whether the analog blocks in Spice respond to changes in the register values programmed in the RTL. This behavior certainly needs to be ensured, because there might be requirement to change the default power-up sequence, such as enabling of converters 4 and 5 before enabling converters 1, 2 and 3 (note that this is a possible scenario as explained in Section 1, unless we have the normal start-up sequence controlled by DDI). In order to achieve this, we programmed the corresponding registers of the digital block to certain values by performing I2C write operation on them. This operation was done with the separate set of lines by the GPIO input.

In the mixed-signal configuration for this scenario, we kept all the five switching converters, housekeeping blocks and the IO blocks in Spice as shown in S.no 2 in Table I above. Though it was a test case to mainly study the effect of changes in the register values on the switching converters' power sequence, we tried one simulation with all the converters in Spice to study the actual effect that is closer to silicon results. We observed that with the appropriate register programming, we could alter the default power-up sequence by first enabling converters 4 and 5 to provide their regulation values, followed by enabling of converters 1, 2 and 3 in that order.

Current measurement: One of the most crucial AMS verification aspect for a PMIC is the measurement of the supply current consumed by the whole chip. This measurement was imperative across different stages of the power up/down sequence, viz., Supply ramp up → LDO enable → IO block enable → Switching converters enable → Switching converters disable → IO block disable → Supply ramp down (which would in turn disable LDO). The measurement was done in order to ensure that the total current consumed by the chip from the supply was within its limits throughout its operation.

Usually during the supply ramp up, the total current consumed by the PMIC would be orders of magnitude more compared to the current consumed during stable operation (active region of the chip). The average supply ramp up current would maintain its level until the LDO is enabled, just after which current consumption would reduce. Enabling of the IO blocks at this point (which makes the chip ready to perform I2C transactions) would not affect the supply current consumption much. From this point, the total supply current consumed by the PMIC would depend on the number of switching converters that are enabled, as more the number of converters enabled more the current consumption (but within the specification). Naturally, during the phase where we turn off the blocks in this sequence in a symmetrical manner, the current consumption gets reduced gradually. We verified the supply current profile of the PMIC in this sequence using Spice simulator for the analog blocks and found the currents to be well within their limits.

Loading effects: The next major but inevitable task was to study the effect of adding a de-rated capacitive load on one of the switching converter outputs. This was done to observe the effect of dynamic variation of the load on the converter's regulation values. Such a load was added by using a variable capacitor that sets the capacitance based on the converter's regulation values during the soft-start (or the ramp-up phase). Upon adding this load, we observed

that the converter 3 (that contains a pre-converter) settled to its near final but not the final value as shown in the yellow dotted region below, which was also the expected behavior.

Figure 7. Converter 2 settling to its near final value with de-rated capacitive load

As evident, such behavior could be verified accurately in AMS. While the whole UVM-based verification setup remained the same from the basic power-on/off test case, we just had to modify the analog control file to include this capacitor. By this method, we isolated such analog block related updates in the setup from the SV testbench thereby avoiding testbench modifications and enabling maximum possible re-use.

IV. Unearthing And Fixing Design Issues In The PMIC With AMS Verification

By performing such AMS verification, we were able to unearth some of the critical bugs in the design, couple of which are listed below. These issues manifested very well in the AMS simulations that we ran by simulating the housekeeping blocks and the switching converters in Spice.

Dampening ripple on the converter output: One of the switching converter (converter 3) was exhibiting dampening ripple on its output along with a ramp, while it was supposed to give an ideal ramp as output (as depicted in figure 2 and figure 7). We caught this through the execution of power on/off test case by considering the loading effects of the converter with a de-rated capacitive load on its output as explained above. Upon catching this issue in AMS verification, corrective actions were taken in the design to get rid of the ripple, and netlist was re-extracted from the schematic. This netlist was integrated with the same UVM-AMS setup, and the power on/off test case was re-executed to verify the smooth ramp behavior with the de-rated capacitance.

Improper current distribution due to vector pin mismatch: The housekeeping block that generates reference voltages and bias currents for the switching converters contained a current carrying vector (bus) pin with an array size of two. This pin which was a <1:2> array was connected to a single (scalar) pin at the receiving end (one of the switching converters), which was a connectivity issue at the PMIC chip-top level. Owing to this, the current distribution at the converter was improper because the total current carried by the bus pin from the housekeeping block was flowing towards only one IP inside the converter, while it was supposed to flow inside two IPs, one from each bit of the bus. This led to reduced voltage level on the converter output. Probing such bias currents in AMS simulations certainly helped in catching issues pertaining to chip-top integration. However, care had to be taken to make sure we saved only the required currents at specific levels of the PMIC, in the interest of runtime and database size. Probing too many currents would not only slowdown the simulation but also consume more disk space. With effective current saving techniques and by using simulator features such as data compression, we could save all voltages and the required currents to verify the intended functionality.

Problems thus identified were fixed in the schematic before taping out the design, thereby ensuring silicon quality and first-pass silicon success. We would also like to mention that upon successful tape-out of the design, our UVM-based AMS verification results were correlated with the actual voltage and current numbers from the silicon and found to be matching. This gave us the required confidence on the verification flow that we followed to ensure the design meets the specifications.

V. CONCLUSION

We could thus perform comprehensive DMS and AMS verification of the Display PMIC with silicon accuracy using the UVM-based mixed-signal verification methodology described above. Seamless re-use of DMS verification environment for AMS simulations helped save huge amount of time in terms of setup, and enabled functional verification of some of the key test metrics with near zero transition time. We could verify the power up and power down sequences, register write and read operations, measure supply currents at various stages of the power sequence, and verify the effects of loading the switching converter with de-rated capacitor. We could unearth some critical design issues in the schematic by performing AMS verification using this UVM environment, and could get them fixed on time before taping out the design. Finally, we could also correlate the simulated data with the actual silicon data and found matching results, thus giving the required confidence on the verification flow/methodology that was followed for the PMIC.

ACKNOWLEDGEMENT

Authors would like to acknowledge the following people from Samsung for providing all the required technical information towards execution of the activity, for the constant support and guidance towards completion of the work on time.

1. Deepa Devendran, Shyam Prasad Venkata Voleti, Vinayak Hegde of Samsung Semiconductor India Research (SSIR) and Divij Kumar (Ex-SSIR)
2. YuBeom Jeon of Samsung Electronics

REFERENCES

[1]. Kiyeong Kim, Hwan-woo Shim, Chulsoon Hwang, "Analysis and Solution for RF Interference Caused by PMIC Noise in Mobile Platforms", IEEE Transactions on Electromagnetic Compatibility, June 2019
[2]. Abdullah Abdulslam, Patrick P. Mercier, "A Symmetric Modified Multilevel Ladder PMIC for Battery-Connected Applications", IEEE Journal of Solid-State Circuits, December 2019
[3]. Chunlei Shi, Brett Walker, Eric Zeisel, Brian Hu, Gene McAllister, "A Highly Integrated Power Management IC for Advanced Mobile Applications", IEEE Custom Integrated Circuits Conference, September 2006
[4]. Nicola Dall'Ora, Sara Vinco, Franco Fummi, "Functionality and Fault Modeling of a DC Motor with Verilog-AMS", IEEE 18th International Conference on Industrial Informatics, July 2020
[5]. Constantina Tsechelidou, Nikolaos Georgoulopoulos, Alkiviadis Hatzopoulos, "Design of a SystemVerilog-Based Sigma-Delta ADC Real Number Model", 22nd Euromicro Conference on Digital System Design, August 2019
[6]. Vijay Kumar, Shrikant Pattar, Yaswanth Chebrolu and Vinayak Hegde, “Harnessing SV-RNM Based Modelling and Simulation Methodology for Verifying a Complex PMIC designed for SSD Applications”, DVCON INDIA 2022
[7]. Nikolaos Georgoulopoulos, Alkiviadis Hatzopoulos, "Real number modeling of a flash ADC using SystemVerilog", Panhellenic Conference on Electronics and Telecommunications, November 2017
[8]. Nikolaos Georgoulopoulos, Alkiviadis Hatzopoulos, "Efficiency evaluation of a SystemVerilog-based real number model", 7th International Conference on Modern Circuits and Systems Technologies, May 2018
[9]. Nikolaos Georgoulopoulos, Athanasios Mekras, Alkiviadis Hatzopoulos, "Design of a SystemVerilog-Based VCO Real Number Model", 8th International Conference on Modern Circuits and Systems Technologies, May 2019
[10]. Spectre Classic Simulator, Spectre APS, Spectre X, Spectre XPS, Spectre FMC Analysis, and Legato Reliability Solution User Guide 23.1
[11]. Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide 22.03