

Mixed Signal Verification of Dynamic Adaptive Power Management in Low Power SoC

Neyaz Khan

Architect - Mixed Signal & Low Power Verification

Cadence Design Systems, Inc.

Plano, TX, USA

nkhan@cadence.com

ABSTRACT

Virtually all modern SoC designs today are mixed-signal in nature. Most systems have to interface their millions of gates, DSPs, memories, and processors to the real world through a display, an antenna, a sensor, a cable or an RF interface. The already complex task of functional verification at the SoC level is getting harder and more time consuming. Up until recently, mixed-signal designs could be decomposed into separate analog and digital functions. Traditionally, digital verification engineers have made assumptions and approximations about the analog components and likewise, the analog designers have made assumptions about the digital behavior. Present day mixed-signal designs have multiple feedback loops with complex system-level interaction between digital and analog components, which is often a rich source of errors. There is a need for an integrated mixed-signal simulation and verification strategy and methodology that can be used to extend advanced verification techniques from the digital verification realm to analog components without compromising speeds needed to verify digital components while preserving the accuracy needed to model and verify analog components.

On an orthogonal plane, the mandate for power reduction is being pursued at every level of IC design for more energy efficient systems. For static power reduction, IC designers are widely deploying power shut-off (PSO) techniques in the design. In applications where PSO is not applicable, power management is often achieved by dynamically scaling the operating frequency and voltage of the target design in real time – a technique known as DVFS (Dynamic Voltage and Frequency Scaling). The verification of DVFS is often a very difficult and delicate task that involves tremendous interaction between the digital and the analog domains. This further increases the complexity of functional verification, which was already a bottle neck, and now becomes even more complex and time-consuming task at the SoC level.

This paper will introduce Digital-centric Mixed Signal (DMS) Verification methodology and provide an overview of how it can enable high performance SoC-level mixed signal verification at digital speeds using Real Number Modeling (RNM) of analog components. Dynamic Power management techniques will be examined in detail. DMS methodology will be applied to an SoC example running Adaptive DVFS as a case study.

1 INTRODUCTION

The complexities involved in SoC level mixed signal verification reveal themselves in a number of ways – ranging from the various languages and disciplines used to the level of abstraction involved in modeling the functionality of complex modern mixed signal SoCs. While the analog components are functionally verified using transient analysis and typically use tools that are based on sparse-matrix based numerical methods, on the digital side functionality is verified using event driven simulation. Both disciplines come together at the SoC level and the task of verifying functional correctness, which was already a daunting task, now gets even more complex and needless to say, much more difficult.

On the digital side, there are advanced verification techniques available to gauge the thoroughness and completeness of the verification effort and require multiple regression runs. With that comes the need for simulation speed and throughput. On the analog side, there is a need for a high-level of accuracy – a single event like a clock edge on the digital side may require numerous computations on the analog engine. This becomes a very severe bottleneck in the MS Verification process. While this accuracy may be required for verification within the analog environment itself, it may not always be necessary for interacting with the digital side. A desirable solution would be to move high frequency events from the analog engine onto the digital engine while preserving the accuracy needed for analog/digital interaction. This forms the basis of a newly proposed Digital Mixed Signal (DMS) Verification Methodology based on Real Number Modeling (RNM) which is presented in this paper in section 2.1.

It's no secret that power management is a key careabout and is emerging as a mandatory requirement in all modern SoCs. While leakage power is well managed by powering down design units that are not functionally needed during the operational cycle of a device [8], there are some key sections of the device that can not be powered down. These

are often very power hungry and need to be addressed. Dynamic power management is a very delicate task whereby the operating voltage and frequency are modulated based on processing needs – see section 3.1. This requires a lot of very complex interaction between the analog and digital components to orchestrate dynamic power management at the SoC level. Needless to say, this is a very difficult task to verify on a digital simulator – so far in the industry there is very little support to verify dynamic power management. This paper addresses the verification of Adaptive Dynamic Voltage and Frequency Scaling (DVFS) as an application of the proposed DMS Verification methodology. See section 5 for details.

This paper will first introduce and propose a DMS Verification Methodology – section 2.2, followed by an introduction to the concept of dynamic power management – section 3.1. It will then apply DMS methodology to an SoC example for verifying Adaptive DVFS – section 5.

2 DIGITAL MIXED SIGNAL VERIFICATION

Digital Mixed Signal (DMS) methodology is based on the use of Real Number Modeling (RNM) to model analog components at the SoC level. With the help of RNM, users can perform verification of their analog or mixed-signal designs using discretely simulated real numbers that involves using only the digital solver. By avoiding traditional, slower analog simulation, intensive verification of mixed signal design can be performed in short period of time.

To meet the verification goals, certain amount of simulation data and data accuracy are required, e.g. a detailed analysis of an RF low noise amplifier requires very high simulation accuracy but a single RF sinusoid period might be sufficient. On the other hand, a pin connectivity check for a large digital block has an extremely low sensitivity towards accuracy but may require a long transient simulation time to cover all sorts of events and states.

Consequently, a long full-chip simulation run using highest level of simulation accuracy would be desirable. The limiting factor in this context is simulation performance. The only practical way around this problem is a hierarchical verification approach that uses different level of design abstractions for different verification goals. Real numbers modeling is an interesting add-on to classical mixed signal verification approaches, like a Verilog and Spice mixed signal simulation or a pure digital modeling of the analog block in the mixed signal design. The extremely high accuracy of traditional analog simulation is traded off for speed while still preserving enough accuracy to enable

highly accurate interaction between digital and analog domain that is required for a full SoC level simulation.

The target audiences for DMS are analog, digital as well as mixed signal engineers seeking high performance mixed signal verification with the ability to:

- Perform high volume, digital-centric nightly regressions tests to verify their mixed signal SoCs
- Verifying Top-level SoCs that have a small to moderate amount of analog in the design

RNM also opens the possibility of linkage with other advanced verification techniques such as metrics driven and assertion-based verification without the difficulty of interfacing to the analog engine or defining new semantics to deal with analog values.

2.1 *Real Number Modeling (RNM) for Digital Mixed Signal Simulation*

The simulation approaches in analog and digital are fundamentally different due to the structure of the underlying equation system to solve. While the digital solver is solving logical expressions in a sequential manner based on triggering events, the analog simulator must solve the entire analog system matrix at every simulation step. Each element in the analog design can have an instantaneous influence on any other element in the matrix and vice versa. Thus, there is not an obvious signal flow in one or the other direction. Time and values are continuous. In digital, time and values are discrete. The solver can apply a well-defined scheme of signal flow and events to solve the system. RNM is a mixed approach borrowing concepts from both domains. The values are modeled as continuous – floating-point (real) numbers to emulate the analog world. However, time is discrete, meaning the real signals change values based on discrete events. In this approach, we apply the signal flow concept, so that the digital engine is able to solve the RNM system without support of the analog solver. This guarantees a high simulation performance that is in the range of a normal digital simulation and orders of magnitudes higher than the analog simulation speed.

Real number modeling capabilities are supported in different standard HDL languages (ref. [7] for details):

- **wreal** ports in Verilog-AMS
- **real** in VHDL
- **real** in SystemVerilog

It is important to note that the real-wire (**wreal**) is defined only in the Verilog-AMS LRM. Thus, a **wreal** can only be used in a Verilog-AMS block. However, it is the digital kernel only that solves the **wreal** system. There are no major performance drawbacks when using these types of Verilog-AMS modules in a digital simulation context.

A detailed understanding of the reference design – in most cases the transistor level circuit – is required for the model creation process. This includes transistor level simulations that are mostly driven from the simulation environment (ADE). Verilog-A models are also frequently used. On the other hand, the model is created for a specific verification purpose with its performance and accuracy requirements. While analog designers mainly own the first skill set, the mixed signal verification engineers understand the verification requirements better. Therefore, a close cooperation between both parties is needed. Figure 1 shows a simple example of RNM using Verilog-AMS wreals.

```

module wrealADC (DOUT, AIN, CK, VDD, VSS);
output [ `Nbits-1:0 ] DOUT;
input CK;
input AIN, VDD, VSS;
wreal AIN, VDD, VSS;
parameter Td=1n;
real PerBit, VL, VH;
integer Dval;

always begin // get dV per bit wrt supply
PerBit = (VDD-VSS) / ((1<<`Nbits)-1);
VL = VSS;
VH = VDD;
@(VDD, VSS); // update if supply changes
end

always @(CK) begin
if (AIN<VL) Dval = 'b0;
else if (AIN>VH) Dval = { `Nbits{1'b1} };
else Dval = (AIN-VSS)/PerBit;
end

assign #(Td/1n) DOUT = Dval;
endmodule

```

Figure 1: Example of RNM using Verilog-AMS wreals

2.2 SoC level DMS Verification Flow

The typical SoC Verification flow involves top-level simulation of components at various levels of abstraction. For example, a verification engineer may need to integrate components from schematics, SystemVerilog, and Verilog (or VHDL)-AMS in a single top-level SoC verification. Figure 2 illustrates a typical MS SoC Verification Env (VE).

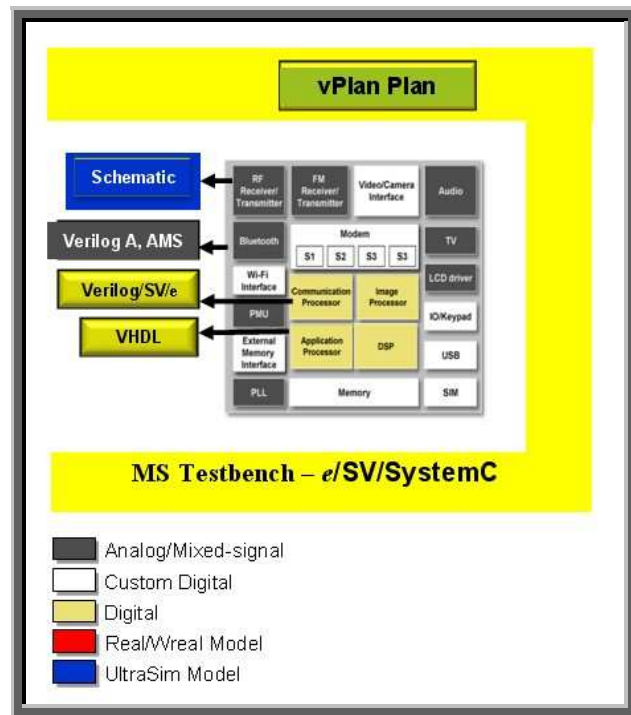


Figure 2: Methodology Stage 1 – Traditional MS-SoC top-level simulation env.

Functional complexity from the analog domain in terms of modes of operation, extensive digital calibration, and architectural algorithms can overwhelm the traditional digital verification methodologies and flow. Simulation at this top-level is extremely costly – both in terms of time and licenses cost, since a significant amount of simulation time for the SoC is spent in the analog engine. Finding a way to reduce the time and expense to verify this SoC, while trading off some accuracy that is not needed at this high level of integration, is extremely valuable. This is the target application of Real Number Modeling. By replacing the analog portions of the SoC with functionally equivalent digital models, which do not require the analog engine, we achieve a significant speed-up in simulation performance. Meanwhile, typical analog simulation problems such as convergence issues are totally eliminated. Figure 3 is a modified version of the above picture with the analog portions of the design replaced with functionally equivalent real Number models.

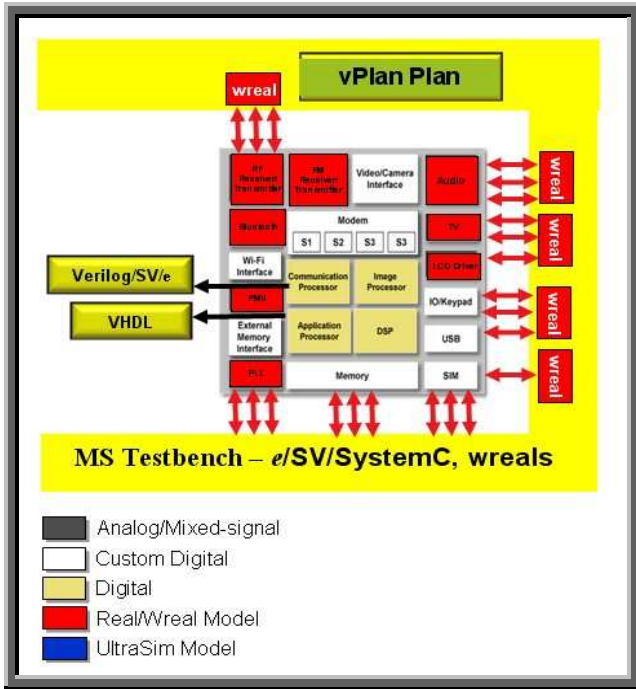


Figure 3: Methodology Stage 2 – DMS-SoC top-level simulation env using RV-Models

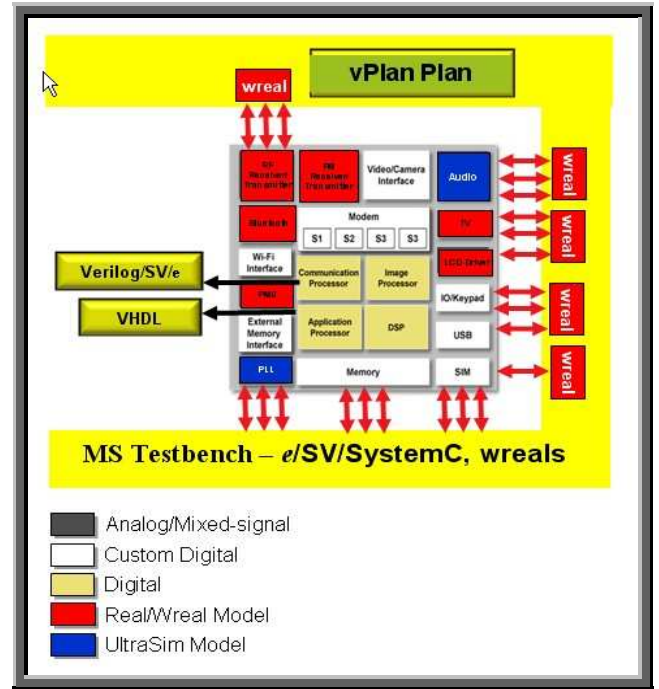


Figure 4: Methodology Stage 3 – DMS-SoC top-level mixed simulation env using RV-Models & detailed analog models

It is important to note that this top level verification strategy illustrated in Figure 3, is not a replacement for detailed block or multiple-block, cluster-level verification with full analog simulation accuracy.

The gain in simulation performance and the reduction in accuracy are highly dependent on the application. There is no general recommendation on what level of abstraction might be useful or not. There are significant advantages of each – simulation speed vs. accuracy must be carefully traded off based on the target application. At the SoC-level verification, there might be rare cases where the RNM based approach does not provide enough accuracy for a particular verification goal, e.g. cross talk of a global net into an analog sub block. In such a case, there is a need to replace RNM models with more detailed analog models – spice/Verilog-A. In the DMS based flow, this can be supported fully as seen in Figure 4, since the RNM models are pin compatible with the more detailed analog models. Other enhancements in the testbench, like coercion of wreal types eliminate the need for any modifications in the VE when swapping analog models with different levels of abstraction. Hence, models coded in wreals, Verilog-A, Verilog-AMS, VHDL-AMS, VHDL-A, Spice can be swapped as needed in the testbench as shown in Figure 4.

Also note that the model verification task – comparing the model against the transistor level reference for each RNM model used in the top-level verification is essential to qualify the overall verification result. This is illustrated in Figure 5, but is beyond the scope of current paper.

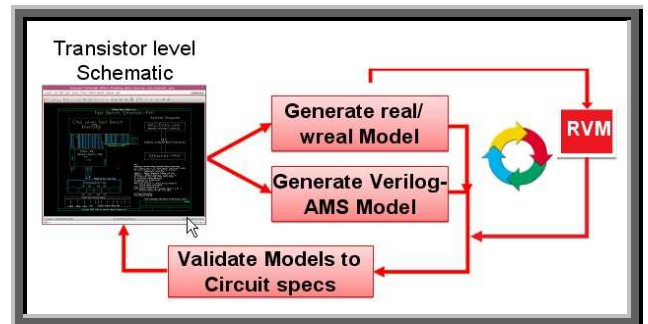


Figure 5: Functional Validation of RV-Models

3 POWER MANAGEMENT

There are two main sources of power dissipation in any CMOS based design - dynamic power is dissipated only when switching, leakage current is permanent and results in a continuous loss [9]

$$Power = P_{switching} + P_{short-circuit} + P_{leakage} + P_{static} \quad [Eq 1]$$

$$P_{switching} = a \cdot f \cdot C_{eff} \cdot V_{dd}^2 \quad [Eq 2]$$

Where a = switching activity, f = clock-freq, C_{eff} = effective capacitance & V_{dd} = supply voltage

Leakage power is well managed by powering down parts of the design when not in use. This problem is well understood in the industry and power aware simulations supported by using either Common Power Format (CPF), [8], or Unified Power Format (UPF), [5].

Dynamic power can be lowered by reducing switching activity and clock frequency which effects performance, and also by reducing capacitance and supply voltage.

3.1 Dynamic Power reduction – DVFS

One of the primary techniques used in the industry for dynamic power reduction is DVFS – Dynamic Voltage and Frequency Scaling [11] which consists primarily of reducing the frequency & voltage of a design – see Figure 6.

The scaling of voltage and frequency is performed in real-time, based on processing needs of the device with the goal of being able to run at the lowest possible frequency and voltage that will support the requirements of the target application.

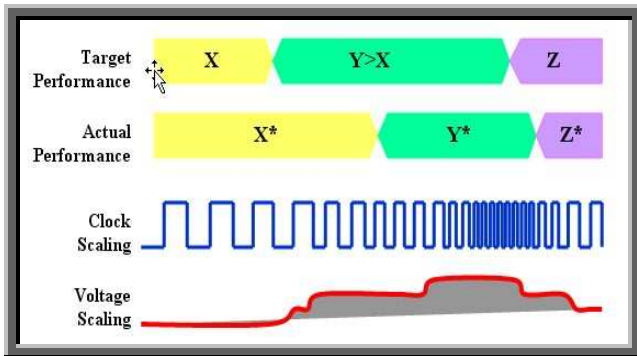


Figure 6: Dynamic Power Management - DVFS

DVFS is used for both:

- Power-saving during off peak processing times, and
- As a protective measure to avoid over heating

3.1.1 Voltage Scaling – Open Loop vs. Adaptive DVFS

Open-loop DVFS is the most commonly used form of DVFS. The operating voltage point or nominal-voltage is pre-determined for the target application and desired operating frequency. The aim is to run the device at the lowest possible voltage while achieving desired performance. The actual clock speed of the device is determined by the PVT (Process, operating Voltage &

junction Temperature). The desired clock-speed is achieved by scaling the voltage to the desired clock-frequency based on statistical data for that process. The operating voltage point for each target frequency is typically stored in look-up tables and used by the power-controller to scale voltage up and down as needed by the application as shown in Figure 7.

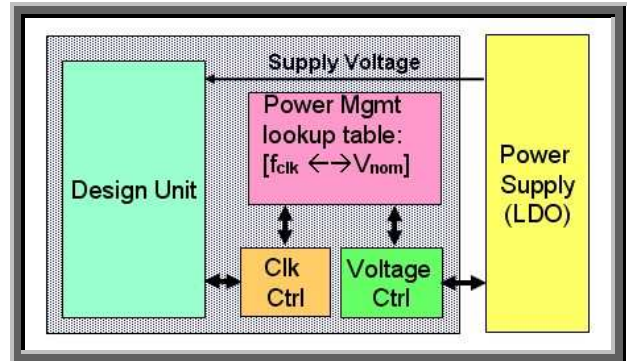


Figure 7: Open-Loop Voltage Scaling

For safety reasons, there are typically large margins assigned to the operating voltage points for each target frequency in the look-up table for open-looped voltage scaling. To achieve the maximum power saving, there is a need to scale the voltages with a much finer granularity. Since the actual operating speed also changes with junction temperature (PVT), there is a constant need to scale voltage to reach optimal power reduction. This is achieved by introducing a feedback loop to the power controller which indicates how fast, or slow a device is actually running based on PVT characteristics. As shown in Figure 8, this task is facilitated by the Hardware Performance Monitor (HPM) which enables closed loop voltage scaling.

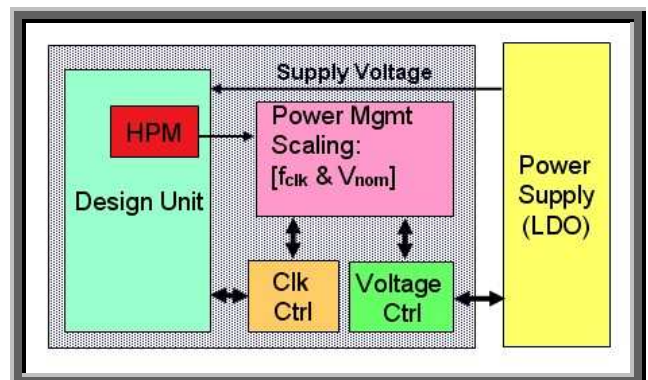


Figure 8: Closed Loop Voltage Scaling

In this paper, closed loop voltage scaling will be simulated using wreal models of the LDO and HPM and frequency

scaling is simulated using wreal models of the VCO. This is described in detail in section 5.4 and section 5.2 .

4 DMS VERIFICATION METHODOLOGY

4.1 Verification Planning

Verification planning is the process of using the spec to define what to check, not looking at the design and defining how to check. By planning what to check, you make sure that you cover all the features that are expected at the SoC level and not just what you've designed into the block. This is a major failure area in MS SoCs in that the analog block is verified under different conditions at the SoC level than were verified at the block level.

Verification planning is one of the key steps to managing the complexities involved in the DMS verification flow. First, it is critical to get all relevant design, verification and implementation teams to agree to what the key low power features and use cases are, and how to verify that they are operating correctly. This translates into the generation of both the system and block level verification environments. These environments need to ensure that they verify all the defined use cases. This translates into coverage, assertion based checking, score-boarding as well as the actual generation of appropriate stimuli for both digital and analog components.

Planning for MS needs to ensure that all the relevant analog components are verified with the well defined and legal operating parameters and operations exercised correctly and thoroughly in the context of DMS Verification. It needs to define the proper modes and the requirements that must be met for each operating mode and transition between modes. Further, it needs to define the exact set of features that need to be verified in each of the designs operating modes.

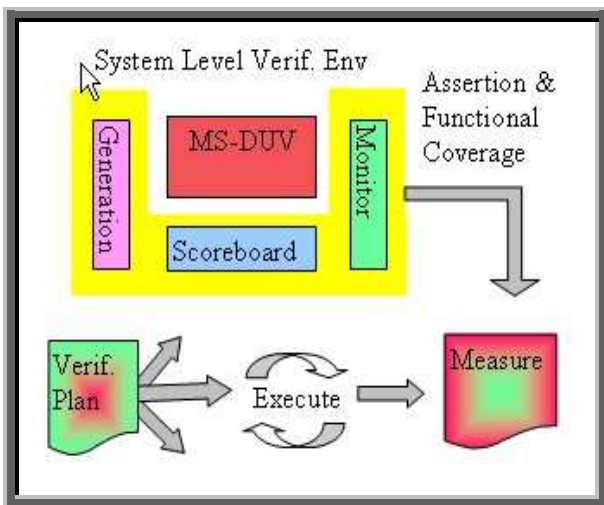


Figure 9: Metrics Driven Verification for MS SoC

The planning process is always important for verification, but for MS it becomes critical. The sheer range of operating parameters and complex interaction of Analog/Digital units requires a clear definition of relevant metrics defined and measured for both Analog & Digital design units – Figure 9. There is a need for a clear definition of what verification closure means in the context of MS operation of the device in the verification plan. This is particularly important as a lot of times, analog components are created and verified in a schematic based environment and often operating margins, modes and details are not captured in a formal spec, which can become a problem at SoC level leading to discrepancies and functional failures. By having it well defined in the verification plan, adequate checkers, monitors, scoreboards and metrics collection units can be created to ensure functional correctness.

4.2 SoC Level MS Verification Flow

As shown in Figure 10, at first, the analog blocks are developed for functional correctness in the traditional analog development environment (ADE). Then RNM based models are created to model analog functionality - see section 2.1 for details on RNM models. These models are then verified for functional correctness in the ADE after which they are handed over to the SoC verification team.

In the SoC verification environment, the RNM models are swapped in place of the analog- blocks, which are pin compatible with the corresponding Spice/Verilog-A/Verilog-AMS/VHDL-A/VHDL-AMS analog models. Since RNM models run on the digital simulation engine, normal regressions can be performed with high throughput and speed. Useful checks and metrics collection can now be performed as part of an overall methodology like OVM [6].

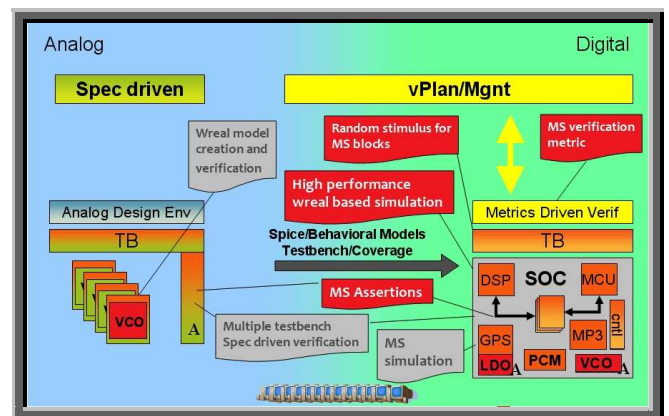


Figure 10: SoC Level DMS Verification

Mixed signal assertions can provide useful checks triggered by either analog or digital events and can accurately check

for relevant interactions between the analog and digital domains. Traditionally, the analog side is black-boxed and checks confined to expected parameters from the analog side but never fully verified inside the analog domain. This has also been a rich source of error in the past. Now, using DMS Verification, these checks can span across the digital/analog boundary and verify complex interactions and sequence of events between the two domains.

Traditionally, there has been very little functional coverage collected from within the analog domain. Now, it is possible to include analog parameters as part of the coverage models when using DMS verification flow – see examples in section 5.6.1 This enables the metrics from analog design units to be included in the verification plan, making it possible to support advanced metrics driven verification methodologies like OVM.

Once all functional verification targets have been met in the SoC verification plan using RNM, some critical tests can now be run swapping out RNM models with the corresponding analog models - Spice/Verilog-A/VHDL-A. No changes to the testbenches are required as the RNM and analog models are pin compatible. Assertion based checks and metrics collection continue to be performed. These targeted tests will take a lot longer to run, and should be carefully chosen, but will serve to increase confidence in the overall verification effort and should be part of the MS verification plan.

5 VERIFICATION OF ADAPTIVE DVFS USING DMS METHODOLOGY

Section 3.1 introduced the concept behind DVFS for Dynamic Power management. Let’s now look at the details of how this is done in a real life SoC – ref Figure 11. Note that the process of Frequency and Voltage Scaling are orthogonal and independent in this application.

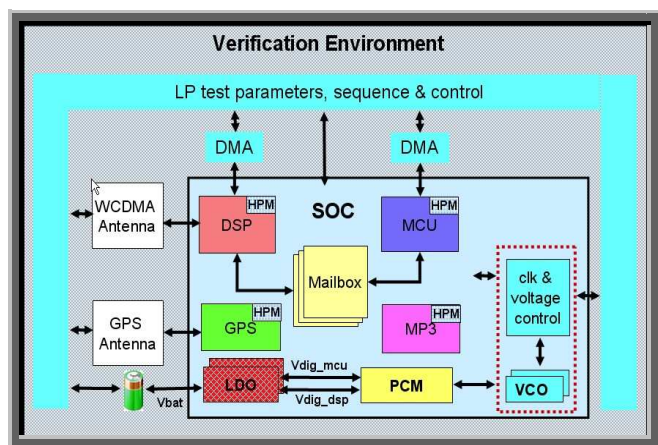


Figure 11: Prototype design for Adaptive Voltage and Frequency Scaling

5.1 SoC Design description

The prototype design consists of the following:

- DSP subsystem – clocked independently: dsp_clk
- MCU subsystem – clocked independently: mcu_clk
- GPS unit
- MP3 unit
- Power Management
- Wreal models: 3 independently controlled Voltage Controlled Oscillators (VCOs) for clock control
- Wreal models: 2 independently controlled Low Dropout Regulators (LDOs) for voltage regulation
- Control unit – clock & voltage

In this design, there are two DMA interfaces that are used to Read and Write independently to a shared Mailbox which is implemented using a dual-port RAM.

There are three independent clocks each controlled by a separate VCO model

- dsp_clk
- mcu_clk
- sys_clk

The dsp_clk & mcu_clk are scaled independently under control from a central clock-controller. The rest of the design runs off sys_clk

A noisy battery voltage is supplied from the verification env and controlled by the test parameters and supplied to the two LDOs:

- mcu_ldo
- dsp_ldo

The two LDOs independently regulate the supplied battery voltage to supply nominal voltages to independent power-domains. There are 5 predefined nominal voltages for each LDO and the outputs are independently scaled under control from the power-controller to supply the regulated voltage to each individual power domain.

5.2 Clock Scaling

The DMA operation from the DSP and the MCU interfaces occur at different rates – see Figure 12. Rate adaptation is performed by independently scaling the individual clocks to match the DMA rates on each interface in order to avoid erroneous DMA read/writes as shown in Figure 13.

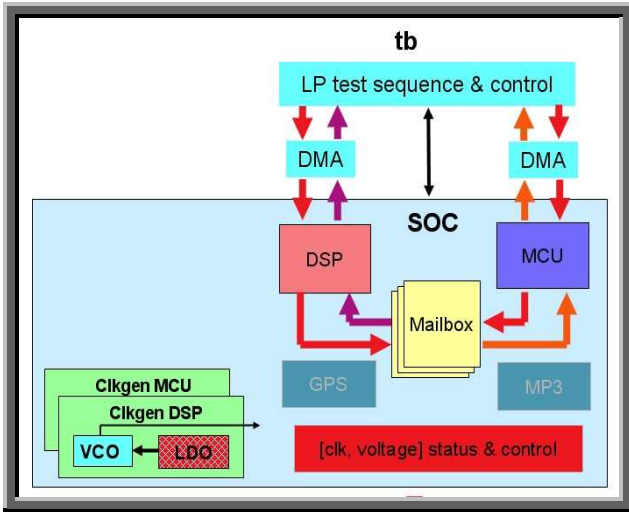


Figure 12: Variable rate DMA

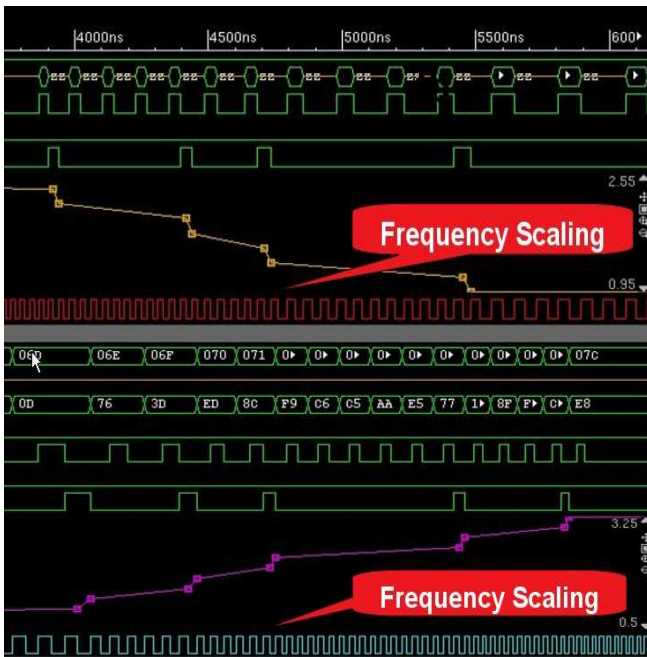


Figure 13: Clock Scaling - Variable Rate DMA Adaptation

Clock scaling is performed by ramping up/down the input voltage to the VCO modeled using wreals as shown in Figure 14. As VCO input voltage changes, the model computes the new clock-frequency which is then applied to the corresponding design. The clock controller modulates the voltage input of each individual VCO independently as needed to reduce the differential between the rates on the two DMA interfaces. In this simple VCO model, the

latency for clock changes is not modeled, but can be easily done so in the VCO model.

```

module vco(vin, clk);
input vin;
output clk;
wreal vin;
reg clk;

parameter real center_freq= 2.5; // in MHZ
parameter real vco_gain = 10; // volts/MHZ

real clk_delay;
real freq;
initial
begin
clk = 0;
freq = center_freq;
clk_delay = 1.0/(2*freq);
end

always @(vin)
begin
freq = center_freq + vco_gain*vin;
clk_delay = 1.0/(2*freq) * 1000;
end

always #clk_delay
clk = ~clk;
endmodule

```

Figure 14: wreal model of VCO

5.3 Controlled Voltage Source

A controlled Voltage Source is needed to emulate the battery-voltage Vbat and to drive voltage into each individual LDO. A noisy voltage-source such as a battery supply that varies over time and also drifts with temperature and use can be created as shown in Figure 15 and Figure 16. Error-conditions like low and dead-battery are also detected and tracked as seen in Figure 16.

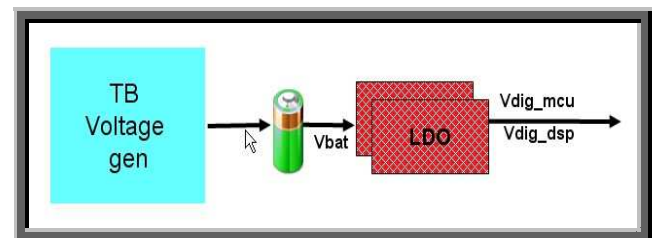


Figure 15: Controlled Voltage Source

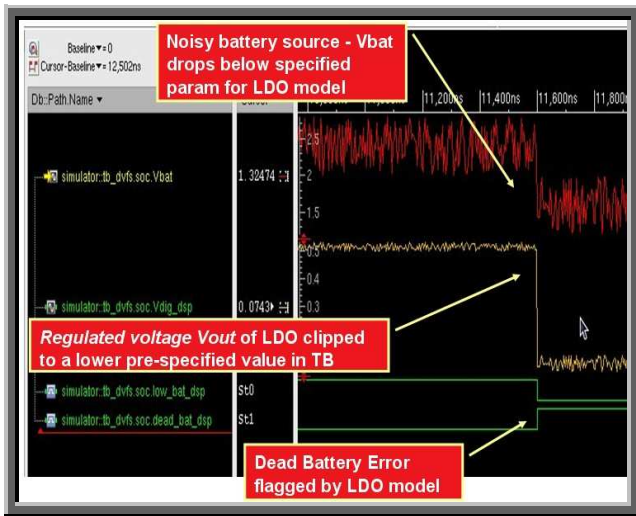


Figure 16: Controlled Voltage Source with error detection

5.4 Adaptive Voltage Scaling

There are dedicated LDOs – ldo_mcu & ldo_dsp, supplying regulated voltage to each power-domain as described in section 5.1. The operating voltages are defined in the wreal models of the LDO and individually configured from the verification env. Voltage transitions are controlled by the power-controller based on estimated processing needs which are task dependent.

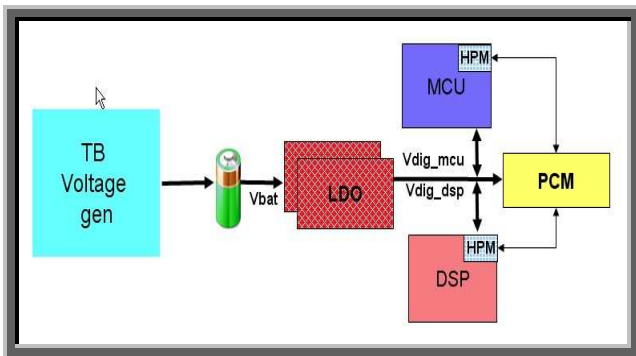


Figure 17: Closed Loop Voltage Scaling

Voltage scaling is orchestrated by controlling the output of the LDO to supply targeted nominal-voltage independently for each power domain. Figure 17 shows how closed-loop voltage scaling is performed on the SoC. The power controller determines the voltage level at which each power domain needs to operate and then fine tunes the supplied

voltage based on feedback from the HPMs – refer to section 3.1.1 for details of closed loop DVFS.

The HPMs are strategically placed inside the SoC and measure the targeted performance for each design unit. They provide feedback to the power-controller to increase, or decrease the operating-voltage in real-time. These are also modeled using wreal. The delay-parameters and update-rates are programmable and controlled from the verification-env.

The results of simulating Adaptive Voltage and Frequency Scaling are illustrated in Figure 18.

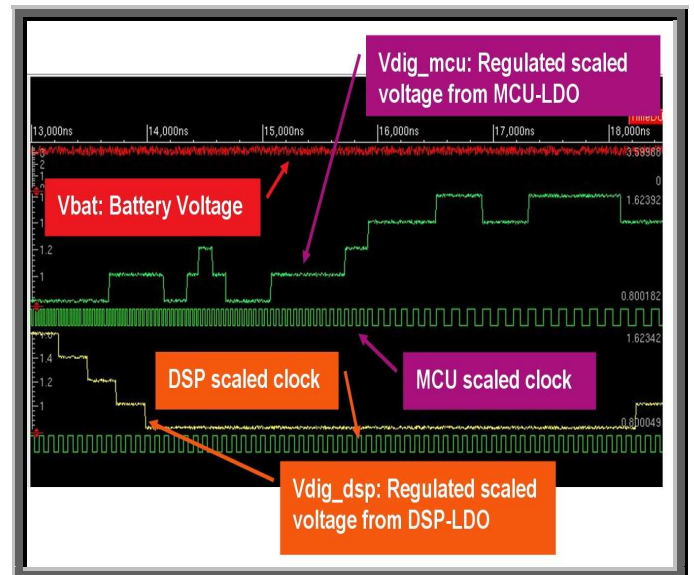


Figure 18: Adaptive Voltage and Freq Scaling

5.5 Voltage Scaling error detection

The fundamental task of any verification exercise is the detection of errors. In this example, randomly generated noise is injected into the regulated output of the LDOs to emulate effects of switching noise and IR drop on voltage of each power domain. This results in glitches, some of which occur close to the transition of nominal-voltages of a given power domain, thus causing faulty voltage transitions. It would be important to detect glitches larger than a specified size and duration. Checkers are put in place to detect these. Anytime these conditions are violated, the entire power domain is corrupted as shown in Figure 19.

Errors are also detected by creating Mixed-Signal assertions that track expected behavior across the digital/analog boundary.

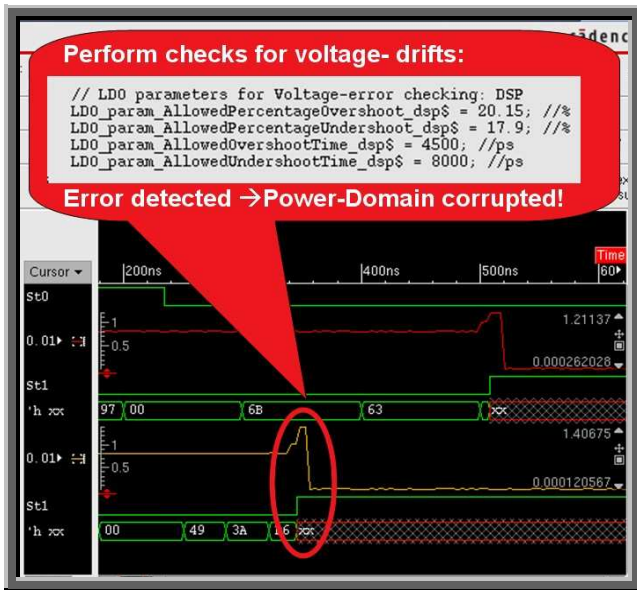


Figure 19: Voltage Scaling error detection

5.6 Verification Closure

Metrics Driven Verification (MDV) is widely used in the industry to measure the quality of a verification effort and to answer the basic questions “am I done verifying my design” [2]. Similarly, Functional Coverage can be used to gauge, and quantitatively measure the quality and completeness of mixed-signal verification. The first step is to be able to collect metrics from Analog elements in the design in addition to metrics collection from the digital side. Coverage Model design that includes Analog elements in the SoC is described in section 5.6.1

5.6.1 DMS Coverage Model Design.

Once the desired features of interest have been extracted from the spec and captured in an executable Verification plan, the next step is to quantify the desired functionality that needs to be tested. This step is typically referred to as Coverage model design – for a detailed analysis and step by step process refer to [3].

The effect of varying Vbat on Voltage regulation by the LDOs is also carefully monitored. Figure 20 shows how the regulated output voltage of MCU-LDO is binned into seven bins:

- On Nom-Voltage = 0.8 V (+/- 10%)
- On Nom-Voltage = 1.0 V (+/- 10%)
- On Nom-Voltage = 1.2 V (+/- 10%)

- On Nom-Voltage = 1.4 V (+/- 10%)
- On Nom-Voltage = 1.6 V (+/- 10%)
- Off Voltage (Powered Down)
- Illegal High Voltage

```

extend cdn_dms_avs_agent_u {
  cover voltage_clk is {
    item Vbat : real = smp.gen_vdd$ using ranges = {
      range([0..1.99], "Dead Battery");
      range([2.0..2.79], "Low Battery");
      range([3.0..3.6], "Good Battery (3.0-3.6V)");
      range([3.7..12.0], "Battery Voltage too High");
    }, illegal = (Vbat > 3.7);

    item Vldo_mcu : real = smp.LDO_op_Vdig_mcu$ using ranges = {
      range([0.7..0.89], "Vnom=0.8 (+/- 10%)");
      range([0.9..1.09], "Vnom=1.0 (+/- 10%)");
      range([1.1..1.29], "Vnom=1.2 (+/- 10%)");
      range([1.3..1.49], "Vnom=1.4 (+/- 10%)");
      range([1.5..1.69], "Vnom=1.6 (+/- 10%)");
      range([0..0.2], "Power ShutOff");
      range([1.7..5.0], "Vnom HIGH <- Error");
    };
  };
};

```

Figure 20: Mixed Signal Coverage Model

Figure 20 shows code for collecting functional-coverage from the wreal models used in the prototype SoC. In this example, the battery voltage Vbat is binned into four categories. Vbat varies over different tests through predetermined ranges and sequences specified in the verification plan and is kept track of to ensure all intended test conditions have been met. Figure 22 and Figure 23 show the cumulative results of the full regression run.

5.6.2 DMS Verification Plan

Section 4.1 goes into the importance and role of a verification plan in the overall verification process. Key metrics and targets for functional completeness of the DMS effort are captured in an executable spec often called the vPlan which attaches itself to the Verification environment. As simulations are run, the coverage information is collected and annotated into the vPlan to give a graphical representation of the percentage coverage you have achieved. The vPlan for some of the analog components in the prototype SoC are shown in Figure 21.

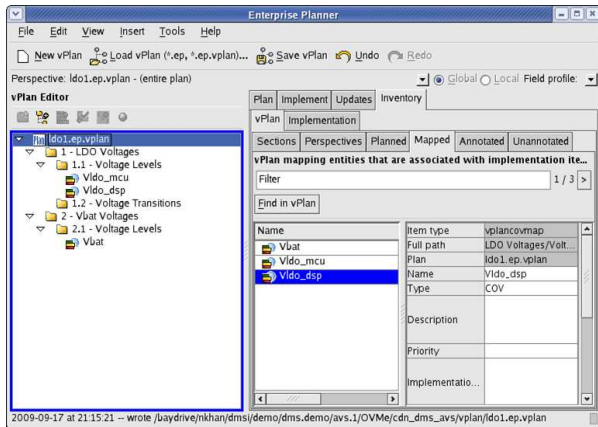


Figure 21: DMS vPlan

5.6.3 Functional Closure of DMS Verification Intent

So what does “closure” really mean in the context of achieving DMS Verification? It would formally be defined as achieving pre-defined verification goals using specified metrics defined in the DMS Verification Plan described in Section 5.6.2.

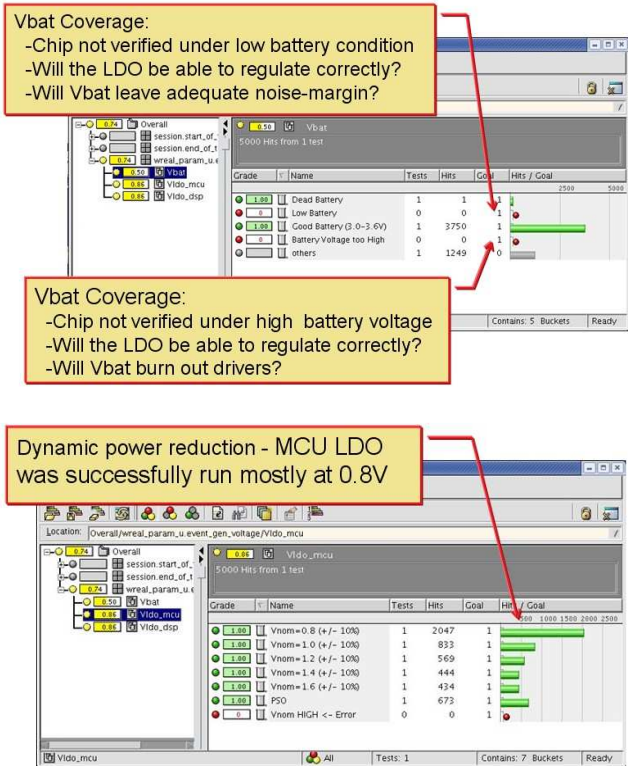


Figure 22: DMS – DVFS Coverage Data

As the device is run through various tests, the output voltages from each LDO are carefully tracked. This gives us

a good measure of the percentage of the time that each power domain in the device is run at lowest possible voltage, which has a direct correlation to the dynamic power actually being conserved (section 3). As seen in Figure 22, the MCU is successfully being run to conserve dynamic power – the higher the coverage data for the lowest nominal-voltage (0.8V), the more power that has been conserved. It is also important to ensure that all possible combinations of LDO voltages have been exercised for all possible legal Vbat values. This can be achieved by creating a cross product of Vbat with Vldo_mcu.

Figure 23 shows the holes analysis run from the vPlan which in turn reveals that the DSP power domain was never run at the highest nominal-voltage of 1.6V. Thus, this part of the plan has not been fully exercised and needs more tests to cover missing condition. Similarly, holes in the verification space are seen in Figure 22. These need to be filled by running adequate incremental tests to achieve functional closure.

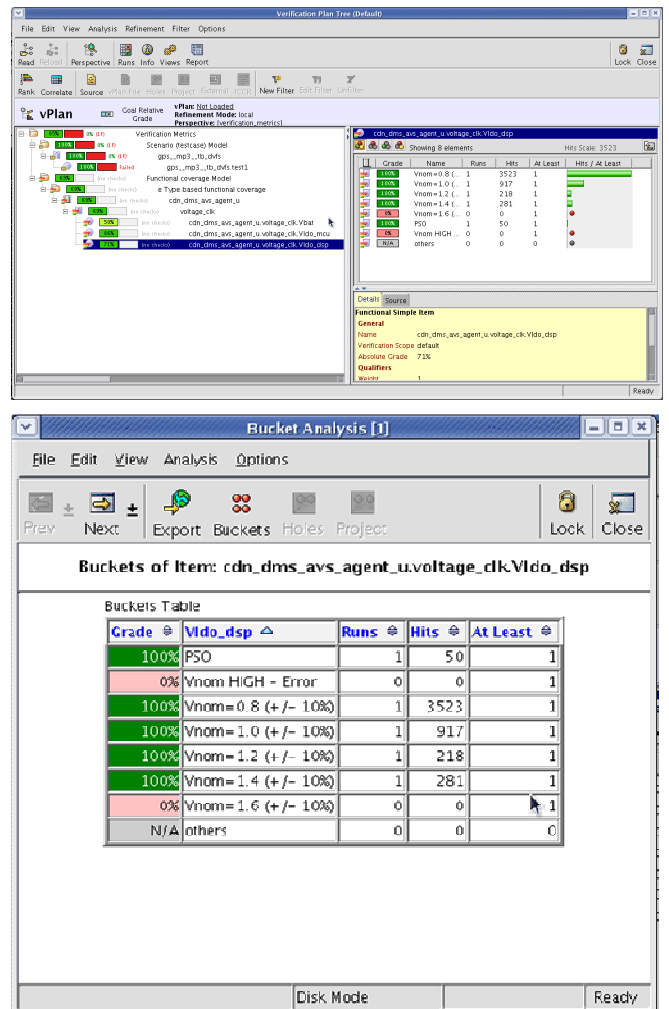


Figure 23: Scaled Metrics in populated vPlan

Note that functional coverage is also collected from MS assertions and are an important gauge of functional completeness.

6 ACKNOWLEDGEMENTS

I wish to acknowledge and thank my friends and colleagues in the Mixed Signal, Functional Verification, Solutions Architecture and Low Power teams at Cadence whose hard work, dedication and knowledge has made this work possible. Some of the background materials have also been borrowed from manuals referred to in the references section.

7 CONCLUSION

The Digital Mixed Signal Methodology provides a reusable, configurable environment to accelerate the verification of Mixed Signal designs. DMS simulations based on Real Number Modeling provide the speed and throughput needed for MS simulation while preserving accuracy needed for analog/digital interaction. This enables advanced Metrics Driven Methodologies to be applied to the full MS SoC.

In this case study, key analog components like LDO, VCO and HPMs are modeled using Verilog-AMS run on the digital engine and provides the framework to run highly accurate Voltage and Frequency Scaling operations for dynamic power management. Complex interactions between the analog and digital domains are precisely executed and measured, errors detected and metrics collected on the full SoC. Error detection and coverage span across digital/analog boundaries to include the full chip. Concepts and techniques of Metrics Driven Verification methodologies are applied with the help of an executable verification plan to achieve functional closure.

The proposed DMS Verification methodology has been fully exercised using a DVFS application to demonstrate its usefulness and how it can be used to extend the verification of analog domain components in an SoC to well established Metrics Driven Verification methodologies.

8 REFERENCES

- [1] <http://www.ovmworld.org>
- [2] Leena Singh, Leonard Drucker, Neyaz Khan, Verification Techniques: A SystemC Based Approach for Successful tapeout, 1st ed. Kluwer Academic Publishers, Norwell, MA, 2004.
- [3] Andrew Piziali, Verification Techniques: Functional Verification Coverage Measurement and Analysis. Kluwer Academic Publishers, Norwell, MA, 2004.
- [4] Common Power Format Language Reference. Version 1.1 – www.si2.org
- [5] Unified Power Format Reference – www.accellera.org
- [6] OVM-based verification flow - http://www.cadence.com/products/fv/pages/ovm_flow.aspx
- [7] Verilog-AMS Real Valued Modeling Guide – IES 8.2, June 2009.
- [8] A Practical Guide to Low-Power Design - <http://www.powerforward.org>
- [9] Neyaz Khan., and William Winkler., 2008. Power Assertions and Coverage for improving quality of Low Power Verification and closure of Power Intent. In DVCON (San Jose, Calif., Feb 19 - 21, 2008).
- [10] Neyaz Khan., and John Decker, J. 2009. OVM Based Methodology for Low Power Designs. In DVCON (San Jose, Calif., Feb 24 - 26, 2009).
- [11] Neyaz Khan., Verification of Advanced Dynamic Power Management in Low Power SoC. In CDNLive! (San Jose, Calif., Oct 14, 2009). http://www.cadence.com/cdnlive/library/Documents/2009/NA/101409%20-%20Track2-1%20Neyaz%20Khan%20-%20Cadence_Final.pdf