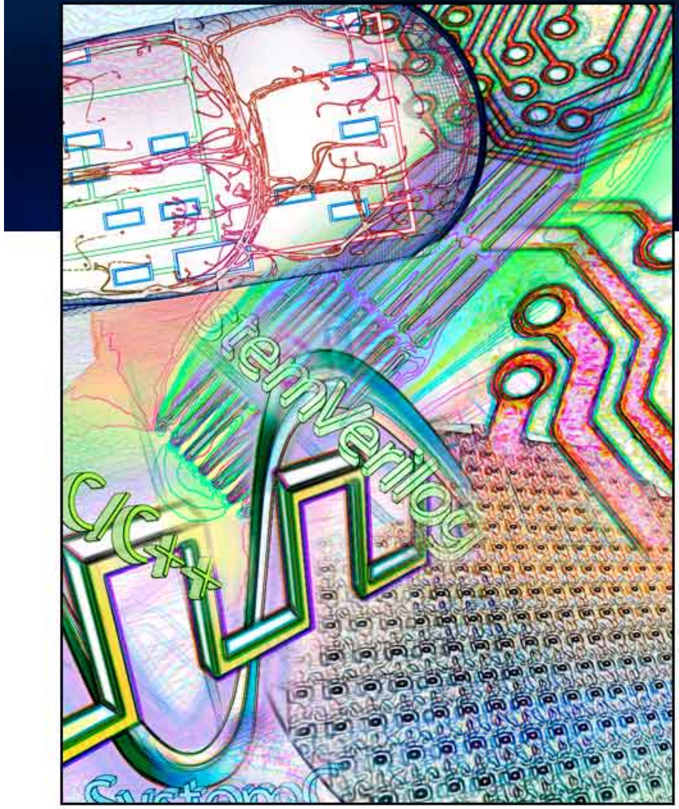
Mixed Signal Verification Transistor to SoC



Martin Vlach

Chief Technologist AMS

July 2014

Mentor Graphics®

Agenda

- AMS Verification Landscape
- Verification vs. Design
- Issues in AMS Verification Modeling
- Summary

AMS VERIFICATION LANDSCAPE

Mixed Signal Verification Challenges

- It is the interactions that kill projects
  - — Simple: Connect "**stuff wrong**"
  - — Complex: Connect the "**wrong stuff**" (algorithms)
  - — Electrical: It's all on the **same silicon**
- Project phases addressed by AMS Verification
  - — Design phase**: RTL *and* schematics**
  - — Power management: **UPF for digital *and* analog**
  - — Post-synthesis phase: **Gates *and* schematics**
  - — Post-layout: **SDF *and* DSPF**
- Human factors
  - — **Understand** each other: Do we "speak" the same tools
  - — **Solve** problems: Advanced debugging and diagnostics
  - — **Integrated flows**: Solving problems, not fighting tools
  - — **Manage** projects: Do we know the state of our project

Four Kinds of AMS Verification

- **Functional Verification**
  - — The task of verifying that digital logic and analog input-output requirements are met

Digital
Analog

- **Parametric Verification**
  - — The task of verifying that numerical requirements are met

Analog
Digital

- **Implementation Verification**
  - — The task of verifying that functional and parametric requirements are met considering all the ways that circuits can “go wrong” in an “analog way”

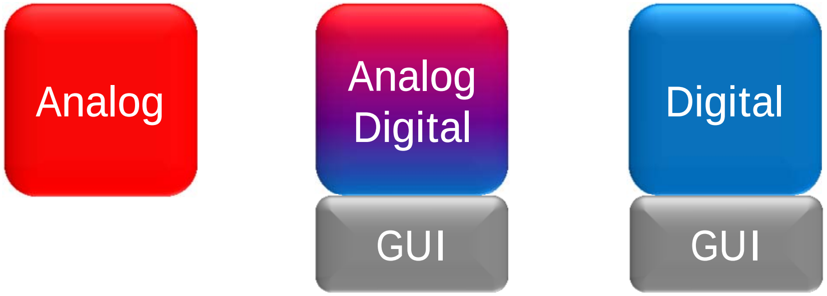
- **Reliability Verification**
  - — The task of verifying that requirements continue to be met as prescribed by the reliability requirements.

Shifting Landscape: Working Together

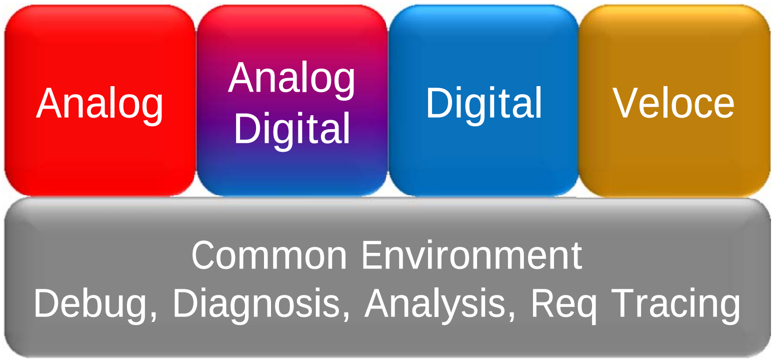
- **Simulation** is the problem
- **Simulators** are the answer

- **AMS Verification** is the problem
- A **common verification strategy** for Digital, Analog and Mixed Blocks
- **Predictable, repeatable, and observable** process
- **Advanced** debugging, diagnosis, analysis, requirements tracing

THE OLD WAY



THE NEW WAY



AMS Verification : Three Solutions

- **Event-Driven** and Real Number Modeling
  - — Fastest, least accurate, big digital best choice.
  - — Questa, Questa ADMS
- **Mixed-Signal** (aka AMS) Modeling
  - — Fast, can be accurate, big analog best choice.
  - — Questa ADMS
- **Digital/SPICE** joined simulation
  - — Slowest, most accurate analog, easiest.
  - — Questa
  - — Eldo, Eldo Premier, ADiT
  - — Analog FastSPICE from BDA acquisition

Putting AMS Together

- **Apart *and* together**
  - Verifying Digital and Analog when apart: building blocks
  - Verifying Digital with Analog together: AMS IP
  - Verifying Digital and Analog integration: SoC
  - Adding power with UPF
- **Testbenches**
  - SystemVerilog with UVM (-MS)
  - SPICE or HDL-AMS
- **Planning and management**
  - Expanding Verification Run Management to AMS
  - Expanding UCDB to AMS
  - Expanding application of ReqTracer to AMS

VERIFICATION
VS.
DESIGN

What Is Verification

- Verification is the science (and art) of asking the question, “What could possibly go wrong?” [Bryon Moyer]
- Verification in a colloquial sense
  - Digital functional verification
  - For analog circuits making sure, using SPICE simulation, that “all process corners have been covered”
  - For mixed signal circuits, verification seems to have the sense of “running a time-domain simulation in a mixed-signal simulator”
- Verification in the formal sense used in Quality Management systems and Functional Safety standards has these objectives:
  - Evidence is provided that the implementation meets the requirements.
  - Traceability is established between hardware requirements, the implementation, and the verification procedures and results.

Verification *vs*. Design

- In the formal sense of standards, design and verification are separate and concurrent processes.
- Design engineers
  - Are concerned with how to meet a specification (perhaps a requirement, or a derived requirement).
  - Think of how everything will “go right”.
  - Do not question requirements and specifications, they implement to them.
- Verification engineers
  - Are concerned with whether a requirement has been met.
  - Think of what can “go wrong”.
  - Worry whether derived requirements correctly fulfill system requirements.

Stakeholders - Technical

- Digital verification engineer
  - Understands verification
  - Uses SV testbenches
  - Relies on RTL delivered from design for digital blocks
  - Uses very high level (TLM) models of analog blocks
    - that reflect digital concerns
    - that are derived from specifications in English
- Analog designer
  - Understands analog design
  - Uses schematics and SPICE
  - Designs to specifications in English
  - Does not trust behavioral models as they do not reflect complete analog interactions observable at SPICE level

Stakeholders - Management

- Management needs
  - Repeatable processes
  - Clarity
  - Predictability
  - Consistency

CONCEPTS IN AMS VERIFICATION MODELING

Modeling Purposes

- Implementation model
  - RTL or gate level for digital.
  - The schematic for analog.
- Physical model
  - Refinement of the design on the way to fabrication. An implementation model together with physical layout (timing, parasitics, etc.)
- Behavioral model
  - Attempts to model behavior even in regions that will not be used.
  - Often used as a reference model.
  - Top-down design and verification.
- Verification model
  - Models behavior only for specific purpose, and uses assertions to identify out-of-spec use of the model.
  - Bottom-up verification.

Model Levels

- **Level 0 - Empty:** A model that literally has no information other than the definition of the interface. Although counterintuitive, empty models can serve an important role as placeholders during integration.
- **Level 1 – Anchored load**: The inputs may present a load to the driving nodes, and the output is fixed at a typical value. The model can include checking code, for example to make sure that correct levels are established by biasing, and can already be useful in simulation to check that inputs stay within design bounds.
- **Level 2 – Feed-through**: The signal levels at inputs are observed and drive a related output value, although the full signal processing functionality of the component is not represented. This may be used for example to ensure that correct supplies are connected.
- **Level 3 – Basic**: At this modeling level, first order functionality of the block is represented at one environment point, with input wiggles being transmitted to related output wiggles.

Model Levels

- **Level 4 – Functional**: The model is fully functional at one environment point, and includes important second order effects. Practically speaking, this is the last level of verification model that is likely to be used in most projects.
- **Level 5 – High Fidelity**: Fully functional model at all environment points. Only very sophisticated teams will find a need for, and especially find beneficial return on investment, at this modeling level. Maintaining such high fidelity models is very expensive, and creating them in the first place and making sure that the model actually reflects what has been built is technically very difficult.

Signal Abstractions in AMS Verification

- Abstractions for modeling a DUT real-life signal
  - Logic
  - Electrical
  - Signal flow
  - Real Number (RN) modeling
  - Event Driven (ED) modeling
- Logic
  - Well understood, digital flows
  - Discrete time, event-driven simulation
- Electrical (special case of conservative)
  - Well understood, analog flows
  - Continuous time, DAEs, need analog kernel
  - Coded in VHDL-AMS, Verilog-AMS, Verilog-A
- Signal flow
  - Not used in verification, only at system/architecture exploration
  - Continuous time, DAEs, need analog kernel

Real Number Signal Abstraction

- Real Number (RN) modeling
  - Similar to signal flow in that information propagates from inputs to outputs; no loading
  - Discrete time, event-driven simulation
  - Signal is represented by one real number
  - Special case of Event Driven modeling
  - Coupled with concepts of X (unknown) and Z (high impedance) values
  - Implementation:
    - `wreal`
      - Verilog-AMS 2.3 with Cadence extensions
      - SV with `wreal` extensions
    - VHDL `real` subtype, possibly resolved, user-defined
    - System Verilog (SV) `real` variable (unresolved single driver)
      - essentially identical to Verilog-AMS without the Cadence extensions
    - SV User Defined Nettype (UDN) with a `real` type, possibly resolved

Event Driven Signal Abstraction

- Event Driven (ED) modeling
  - ED modeling is a proper superset of RN modeling
  - Signal is represented by multiple pieces of information, e.g.
    - voltage/current/impedance
    - frequency/phase/dutycycle
  - Implementation:
    - VHDL: composite type, usually `record`, possibly resolved
    - SV: aggregate type, usually `struct`, possibly resolved, using the User Defined Nettypes (UDNs) construct

Connectivity

- Mixed signal design
  - Where digital and analog flows interact
  - Not necessarily two simulation engines
- Mixed (digital) language (aka Mixed design in digital idiom)
  - SV and VHDL, often for SV testbench and connectivity and VHDL Design Under Test (DUT)
- Mixed abstraction
  - Logic and electrical for mixed engine
  - Logic and RN in digital engine
  - Logic/electrical/RN/ED in the most general case
- Mixed kernel simulation
  - Implies mixed abstraction: commonly electrical/logic or electrical/RN
- Connectivity: putting models together:
  - Abstract connectivity (no representation known)
  - Concrete connectivity (representation explicitly specified)
  - Structure from schematic with netlisters

Mixed Abstraction Issues

- May arise in digital-only kernel simulations, always arise in mixed-kernel
- Analog abstraction conversion
  - Boundary element insertion is subject to discussion related to accuracy and simulation speed tradeoff: where and how many?
- `electrical - logic`
- `electrical - wreal`
- `wreal - logic`
- `electrical - wreal - logic`
- Multiple types in general, including all abstractions and languages

Major AMS Flows

- Testbenches with UVM for MS
  - Stimulus: driving RN and ED signals from SV testbench
  - Monitoring: reading ED and RN model values from SV testbench
- Low Power with UPF for MS
  - Connecting the UPF power/ground with the analog world
    - Explicit supplies, either in SPICE or as RN or ED
  - Abstraction conversion (boundary element insertion) in the presence of multiple power domain
    - With level shifters, isolation cells, and retention cells inserted
    - With direction signal connection between power domains
- Model Verification
  - Not model equivalence checking as understood in digital
  - Making sure the verification model corresponds to the transistor implementation
  - Use the same testbench to drive SPICE and RN/ED models

Languages

Language	Strengths
Simulink (simulator)	Architecture, signal flow
SystemVerilog(SV)	Large complex SoCs, UVM testbench, DUTs, Emulation
Verilog	Pre-SV, basis for Verilog-AMS
VHDL	System through semiconductor, early HDL
SystemC	System level
Verilog-A	Semiconductor, PDKs, behavioral analog (no digital)
Verilog-AMS	Digital-analog behavioral modeling, semiconductor
VHDL-AMS	System through semiconductor, early HDL-AMS
SystemC-AMS	System level
SV-AMS	Early stages of standard work
SPICE	Transistor level

Practical problems in AMS Verification

- Have a transistor implementation of an analog IP block
- Have a model of that implementation (2k-3k digital pins)
- Convince me that the model(s) faithfully represent(s) the circuit
  - The full specifications of the circuit is modeled
  - For the purpose to which the circuit will be put
    - Do verification "in situ"
  - That if I misuse the model (connect it incorrectly, connect it in the wrong environment) that I will
    - Find out that there is a bug
    - Find out what the bug is
- Have an SoC testbech that uses the model
  - Convince me that the testbench tests all the requirements

Verification Modeling Best Practices

- Executable specifications (reference and verification models) are used in addition to English.
- Modeling technologies and styles are consistent with the verification problem.
- AMS verification modeling of mixed signal and analog blocks includes both the verification engineer and the analog designer.
- Continuous modeling effort is an integral part of the verification process, not an afterthought or a one-time activity.
- Assertion Based Verification (ABV) is used in the MS models.
- Model verification is an integral part of the regression testing environment.
- Formalized testbenches (SV?) are used in verifying the model against the SPICE implementation.

SUMMARY

Summary

- Mixed signal verification is more than running a mixed-signal simulations.
- Mixed signal verification methodology often requires investment in mixed signal modeling.
  - To catch bugs early
  - To cope with specification changes
- Modeling of AMS blocks for SoC verification does not have a one-size-fits-all answer.
  - AMS modeling is difficult in the first place
  - Knowing what not to model is harder
  - Knowing whether everything that needs to be models in fact is modeled is hardest - analog coverage

Mentor Graphics®

www.mentor.com