Analog Mixed-Signal (AMS) Verification Challenges

Vijay Gor

OmniPhy
www.omniphysemi.com
What we do at OmniPhy

**Multi-Media**
- HDMI 2.0 TX/RX
- HDMI Tx Controller
- HDMI Rx Controller
- MHL 3.0 PHY
- HDCP

**Ethernet**
- 10/100 Ethernet, w/ EEE
- 10/100/1000 Ethernet
- Automotive Ethernet
- XAUI PHY

**USB**
- USB 3.0 PHY
- USB 2.0 PHY
- USB 3.0 Device Controller

**Networking**
- Super Computing
- 16-28Gb/s CEI
- 56Gbps PAM

**MIPI**
- SLIMbus Master/Slave controller
- M-PHY 3.0

**PCle**
- PCIe 1.0 PHY
- PCIe 2.0 PHY
- PCIe 4.0 PHY

**Storage**
- SATA 3.0 PHY
- SD 3.0 Host/Device Controller

**DDR**
- DDR4 PHY
- DDR3 PHY
- LPDDR 2/3 PHY
Why AMS

- To verify Mixed Signal designs

- SPICE-based simulation provides the accuracy needed for the analog design, but is too slow to handle the system level verification

- Digital simulation provides the necessary speed to simulate the digital design, but fails when dealing with analog parts

- AMS solves these problems partly, but introduces co-simulation challenges

- Common Issues in Mixed Signal IP’s
  - Inverted Polarity
  - Pins connected to the wrong power domains
  - Incorrect bus order
AMS Verification Challenges

- Simulation Time
  - 1us of simulation time takes 7 hours of real time
  - Extra efforts for initial environment setup

- Modelling
  - Conventional models will not help
  - Need Analog Behavioral Model (Verilog-A/VHDL-A/SystemC-AMS) or
  - Real Number Model

- Randomization of analog signals
Challenge 1 – Simulation Time

- Huge simulation time
  - 1us of simulation time takes 7 hours of real time
    - Observed on a design consisting of PLL, ADC, DAC, LPF, Squelch, etc.
    - Using Cadence’s Virtuoso Multi Mode Simulator
  - Almost impossible to run all test cases
  - Extra efforts for initial environment setup

- To improve simulation time, we have to compromise with accuracy

Source: mixed signal challenges whitepaper from Cadence
Performance Trade-offs

Source: mixed signal challenges whitepaper from Cadence
Challenge 2 - Modelling

- No standard guidelines for type of model to use at different abstract levels
- Conventional pure digital models will not help
- Need to develop Analog Behavioral Models
  - Using Verilog-A, VHDL-AMS, etc. OR
  - Real Number Model using “real” in Verilog
    - Value is continuous but the time is discrete
    - Less accurate but high performance for simulation
- Need expertise to hand code analog models
  - Analog engineers are best suited to do this but most of them lack the programming skills or few are familiar with the language
  - Digital engineers have that familiarity but know less about the analog circuits
- Common mistakes in hand-coded behavioral models
  - Clocks are not derived out of correct source
  - Missing or incorrectly driven status signals
  - Incorrect wait time
- Equivalence checking between AMS Model v/s Transistor circuit
Modelling Trade-offs

- AMS Modelling involves trade-off between simulation time and accuracy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>VHDL-AMS</th>
<th>Verilog-A</th>
<th>SystemC-AMS</th>
<th>SPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Med - High</td>
<td>Med - High</td>
<td>Med - High</td>
<td>High</td>
</tr>
<tr>
<td>Abstraction Level</td>
<td>Any abstraction level</td>
<td>Any abstraction level</td>
<td>Any abstraction level</td>
<td>Transistor / Netlist level</td>
</tr>
<tr>
<td>Time to model</td>
<td>Less to Medium</td>
<td>Less to Medium</td>
<td>Less</td>
<td>High</td>
</tr>
<tr>
<td>Simulation Run Time</td>
<td>Less</td>
<td>Less</td>
<td>Medium - High</td>
<td>High</td>
</tr>
<tr>
<td>Tools required</td>
<td>Simulator</td>
<td>Simulator</td>
<td>AMS libs with GCC/ Simulators</td>
<td>Simulators</td>
</tr>
<tr>
<td>Support</td>
<td>Available</td>
<td>Available</td>
<td>Less</td>
<td>Available</td>
</tr>
</tbody>
</table>

Table 1: Modeling approaches and performance trade-off.

Source: AMS Modelling approaches – Design & Reuse
Digital blocks (PD and Modulo-N Div) can be modelled in Verilog / VHDL
Analog blocks (Low Pass Filter and VCO) can be modelled in SystemC AMS / Verilog-A / VHDL-A / Spice
Model all blocks in SystemC AMS/Verilog-A/VHDL-A
Challenge 3 - Randomization

- Randomization of analog signals
  - Analog signals are represented by “real” variables in AMS models
  - “real” variables cannot be randomized

- Possible workaround
  - Use SV constraint for integers
  - Convert constrained integers to real
To Summarize

- AMS provides a good opportunity to simulate digital and analog blocks together
- But throws out its own challenges of co-simulation
- Simulation Speed is the primary challenge
- We can overcome this challenge by Modelling
- But modeling has its own challenges
- To overcome these challenges we need
  - Extra Resources
  - Expertize in AMS language(s)
  - Co-ordinated effort between Digital and Analog engineers
  - Proper planning to trade-off simulation time v/s accuracy v/s model development time
Thank You

Answers?