Advanced Digital-Centric Mixed-Signal Methodology

Methodology for Analog and Mixed-Signal Model Integration and Progression inside a Mature Digital UVM Testbench

Michael Kontz
David Lacey
Peter Maroni
What is Digital-Centric Mixed Signal?

*Bringing mixed signal verification to the powerful functional verification world*

- Existing full featured testbench
- Coverage Driven Verification
- Random stimulus
- Automated checking
- UVM architecture
- Full suite of existing tests
- Automated 24x7 regressions
Analog Circuit Verification

- Many small subcircuit simulation environments
- Simple digital controls provide required stimulus
- Manual checking
- Still requires full gate models for system sims

High investment costs
Our Decision…

**Circuit development needs**

- Complex algorithms involving more digital logic interaction
- Historical analog / digital bugs
- Unknowns of new technology node

**Where do we invest the effort?**

- Bring UVM to the analog world?
- Bring analog to the discrete testbench?
Our Motivation

Low investment cost / high ROI

- Greater value (high return) using existing infrastructure
- No changes (low investment) needed to digital testbench

Full power of existing testbench

- Full suite of tests
- Advanced DV techniques in place

Tunable to the exact desired circuits

- Focus on the larger algorithm around the circuits
Target - PLT UVM Testbench

HALF-0

Sequences
  Training Traffic Channel

Register Bus UVC

Core UVC

Link UVC
  Checkers Register Model

Clock UVC

Channel UVC

Link DUT

DUT RTL Parameters

Swap in analog models here
# Models

<table>
<thead>
<tr>
<th>Type</th>
<th>Author</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| High Level Discrete | V      | • Enable digital testbenches  
• Bulk high level functional verification                                                                                               |
| Low Level Discrete  | A      | • Required for all low level cells  
• Enables gate models in digital testbench  
• Digital connectivity                                                                                                                   |
| Abstracted Models   | A      | • Balance model accuracy with testing objectives (RNM, VerilogA, VerilogAMS)  
• Example: equalization (de-emphasis) on analog front ends                                                                                  |
| Analog              | A      | • Full analog circuit provides highest accuracy  
• Process/skew aware  
• Targeted only for specific subcircuits.                                                                                                  |
Getting started

Build flow
- Mature AMS tool flows
- Cadence irun supports AMS

Model swapping
- Verilog Config Block (beh/RNM/VerilogA)
- AMS config card (SPICE)

Challenges
- Maturity of mixed-signal tools low (Verilog 2009/SV)
- Full control of analog settings from random TB
Addressing Performance

**Speed Improvements**
- Shorten reset period
- Parameterization limits amount of analog models
- Adjust to testing focus
- Traditional approaches
  - Backdoor CSRs
  - Tune analog solver parameters
  - Alternate models (RNMs, etc.)

8x faster simulation time
UVM Testbench is ready! Where are my Models?

- The PLT and PHY designed in parallel.
- PHY circuits designed bottoms up meaning no complete design until late into the project.

Solution: Model Progression Plan
Model Progression Plan

- Project long framework for upgrading behavioral models with completed schematic designs (gate models) over time.
- Pieces of the gate model can then be further upgraded with *MS or Analog models.
- The order of model delivery into this framework.
Generic

- Worked with analog microarchitect to define the long term high level hierarchy of the overall design.
- Generic behavioral models for major subblocks.
- Enables UVM digital testbench before physical design begins.
- Wrappers and SystemVerilog interfaces hide internal upgrades to come.
Hybrid

- Mixture of behavioral and gate models.
- Wrappers use Verilog `parameter` or `define` to swap between model type.
- Extra logic needed to bring generic behavioral up to par with real gate design.
- Behavioral + gate combos dictated by model delivery order.

Key:  
- **Common**
- **Wrapper**
- **Behav**
- **Gate**
- ***MS**
Hybrid-*MS

- With gate model in place, MS models can now be used.

```verbatim
cfg phy_vco_vams;
design worklib.top;
instance top.half0.phy.rx_pad.pll.vco use worklib.mikes_vco:rnm;
endconfig
```

**Additional Wrapper Benefits:**

- Keep non-logic discipline contained
- Convert to *MS friendly Verilog constructs

**Models 1:1 with schematic for easy swapping**

**Key:**

- **Common**
- **Wrapper**
- **Behav**
- **Gate**
- ***MS**
Hybrid+SPICE

- SPICE models can be added as well.

```plaintext
include "vco.scs"
amsd {
    // match ports between schematic and module using port names
    portmap subckt=vco autobus=yes reffile="vco.v" porttype=name
    config inst="top.half0.phy.rxpad_pll.vco" use=spice
}
```
• Entire model is complete gate level extract from schematics.
• Mixed signal models can be swapped in again for final testing.
### Roles

<table>
<thead>
<tr>
<th>Verification Engineers</th>
<th>Analog Engineers</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Create and maintain testbenches</td>
<td>• Develop low-level analog circuits</td>
</tr>
<tr>
<td>• All infrastructure tooling</td>
<td>• Develop gate model hierarchy</td>
</tr>
<tr>
<td>• Create high-level behavioral models</td>
<td>• Create and verify Mixed Signal models</td>
</tr>
<tr>
<td>• Debug and triage</td>
<td></td>
</tr>
</tbody>
</table>
Results

- **Small Investment**
  - Low cost to add complete mixed signal capability to mature UVM testbench

- **Real Benefits**
  - Added models reaped benefits of more complete stimulus and checking

- **Model Progression Plan Worked**
  - Enabled testbench well ahead of schematics
  - Swapping gate/*MS models later was easy

- **Cross Discipline Understanding**
  - Grew team’s cross discipline knowledge and teamwork
**Sim Performance**

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Serializer 16 Subcircuits</th>
<th>Serializer All Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete Behavioral</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Real Number Model</td>
<td>2.5</td>
<td>n/a</td>
</tr>
<tr>
<td>Verilog-A</td>
<td>5</td>
<td>n/a</td>
</tr>
<tr>
<td>Fast-SPICE*</td>
<td>14</td>
<td>501 (8 hours)</td>
</tr>
<tr>
<td>SPICE*</td>
<td>40</td>
<td>4740 (79 hours)</td>
</tr>
</tbody>
</table>

* Huge variability depending on accuracy and waveform settings (range: 1 hour to weeks)
Future Plans

More Real-Number Models

• Enable full length link training simulation.
• Target PLT+PHY holistic items like automatic equalization.

Save and restore capabilities

• Restore mixed signal sim after lengthy reset and calibration with different stimulus.
## Conclusions

| Complete flexibility with minimal investment | Languages and tools are ready  
Only incremental effort required – high ROI |
| Mixed environment not equal capability to digital | Mixed signal tools are still maturing  
Shield MS models from advanced RTL constructs |
| Model Progression plan achieves earlier verification | Schematic designs and *MS models added to UVM testbench as soon as they’re available  
Plan == Teamwork |
Questions

Advanced Digital-Centric Mixed-Signal Methodology

Methodology for Analog and Mixed-Signal Model Integration and Progression inside a Mature Digital UVM Testbench

Michael Kontz
David Lacey
Peter Maroni