Taking RNM Model to the Next Level: Power-Aware Verification of Mixed-Signal Designs

Subin Thykkoottathil
Nagesh Ranganath
Santosh Singh
Jakub Dudek
Nimay Shah
Agenda

- Introduction
- Low power verification challenges
- Traditional verification approach
- RNM-based approach
- Bugs found
- Conclusion
Introduction

• Every new generation of electronic devices requires higher performance and more sophisticated power management.

• This poses an exponentially growing verification challenge, especially in the mixed-signal domain.
LOW POWER VERIFICATION CHALLENGES
Mixed Signal Interactions

- Interactions of digital controllers with analog blocks such as
  - References, LDOs, Voltage monitors, Analog Power Switches
  - Power-on circuitry, Clock sources, etc.
- Isolation/Clamp values of the control signals during power up and other power modes must be carefully selected and verified
Complex Power Mode Transitions
TRADITIONAL VERIFICATION APPROACH
Digital simulations

- CPF/UPF power aware simulation
- Simple non-pin compatible Verilog model for analog
  - Effects of power transitions on analog is not verified thoroughly
  - Isolations between analog power domain is not verified
Reliance on Cosim

• Cosim for signing of Low power (mixed signal) verification
  – Slower Bring-up of Cosim Verification Environment as compared to digital DV
  – Large turn-around time of cosim runs
  – Coverage closure can be achieved only after merging digital and cosim dumps
  – Finding bugs towards the end (one of the long pole in all previous projects)
RNM-BASED APPROACH
Simplified Design

Chip Top

Analog
- Reference
- POR
- Voltage monitors
- LDO
- Power switch
- OSC
- PLL
- ADC

Digital
- AON
- PGT
- ROM/SRAM

External supply

Inspiration

Control signals

Supply
Verification Environment

Chip Top

Analog (Power aware RNM Models)
- Reference
- LDO
  - Voltage monitors
- POR

Digital
- OSC
- PLL
- ADC
- AON
- PGT
- ROM/ SRAM

UVM Supply Agent
(random slope, amplitude, noise, waveform)

External supply

Supply

Control signals

External supply

UVM interrupt Agent
(random delay interrupt source synchronized to C counter part to wakeup)

C Tests
(Testcase sequences with randomness)

(UVM Supply Agent (random slope, amplitude, noise, waveform))
Real number modelling

... real out_int;
assign out_int = limit(out_int + I_lDo * step_size/Ci, .min(0.0), .max(vref));

... function real limit (input real in, input real min, input real max);
limit = (in > max) ? max : (in < min) ? min : in;
endfunction
Power awareness

assign out_int = limit(out_int + I_ldo * step_size/Cl, .min(0.0), .max(vref));

assign out = (valid_inputs && bias_good && pwr_good) ?
  (pdb === 1'b1)? out_int : `wrealZState ) : `wrealXState;

function real limit (input real in, input real min, input real max);
  limit = (in > max) ? max : (in < min) ? min : in;
endfunction

inout vdd;
cds_rnm_pkg::wrealsum vdd;
always @(*)
if ((vdd !== `wrealXState && vdd !== `wrealZState) && (vdd >= vddl_min)
  && (vdd <= vddl_max) && !vss && !vsub )
  pwr_good <= 1'b1;
else
  pwr_good <= 1'b0;
Netlisting

• Standard cell Netlisting, no modelling required
Model validation
RNM Flow

- SV RNM stored in Virtuoso as “SystemVerilog” view
  - Usage of Cadence built-in nettypes from `cds_rnm_pkg::wrealsum`
- RNM Check & Save
  - Pin matching hierarchical checks
- Analog Verilog Netlist
  - Pin compatibility ensured during netlist process
  - Netlist using config views with Cadence SV Plugin
- Chip Top Connectivity
  - Updated through scripting, no human interaction
  - All digital islands exposed DMS hierarchy
- Compilation/Elaboration
  - Discipline resolution, datatype coercion and connect module insertion solved In order during elaboration.
- Simulation
  - Validation flow against Schematic
  - Model usage in DMS environment
Integration of RNM with digital RTL

• Using datatype coercion
  – Connecting the models to the rest of design/testbench without worrying about the interconnects and their types.
Driving supply from Testbench

• Supply agent generating glitches, over-voltage and under-voltage scenarios
Other techniques used

- Actual supply from the RNM model is used as a shutoff condition for a power domain.
  - MORE accurate behavior of power gating (timing wise).
  - Verification of analog logic from the power switch digital control to the actual output of the switch (LDOs in many cases).

```plaintext
# Core domains
if {$env(SIMULATION) == "false"} {
  create_power_domain -name PD_VDDMEM -instances $Mem_Insts
  -shutoff_condition "${$dig_scope}pgsw_pd"
}
else {
  create_power_domain -name PD_VDDMEM -instances $Mem_Insts
  -shutoff_condition "${$dig_scope}VDDMEM"
}
```
BUGS FOUND
Wrong isolation values

- Wrong isolation values leading to POR or wake from sleep failures.
Analog block sequencing

• Wrong analog block sequencing leading to power mode transition failures.
  – PLL not working after a power mode wakeup
  – Digital power failure on one of the power mode transition.
ADVANTAGES IN THE NEW FLOW
Exhaustive Isolation Verification

- Power aware RNM model are really powerful to catch most of isolation issues in functional simulation. Including the analog to analog isolation.
  - Wrong isolation values on TRIMs, LDO controller delays etc.
- Isolation assertion generated from spec (isolation spreadsheet)
- Isolation toggle coverage
  - Auto-generated from the isolation spreadsheet
Coverage Closure without Cosims

- Achieved 100% code and functional coverage

<table>
<thead>
<tr>
<th>Name</th>
<th>Overall</th>
<th>Overall Covered</th>
<th>Block</th>
<th>Branch</th>
<th>Expression</th>
<th>Toggle</th>
<th>FSM</th>
<th>Transition</th>
<th>Functional</th>
<th>Assertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>aon_subsys</td>
<td>100%</td>
<td>4355 / 4355 (100%)</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>n/a</td>
</tr>
<tr>
<td>pmuPGT</td>
<td>100%</td>
<td>6748 / 6748 (100%)</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>n/a</td>
</tr>
<tr>
<td>u_A_D_ASSERTIONS</td>
<td>100%</td>
<td>9 / 9 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_A_D_ISO_ASSERTIONS</td>
<td>100%</td>
<td>156 / 156 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_A_D_POR_ASSERTIONS</td>
<td>100%</td>
<td>93 / 93 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_A_D_scan_latch_assertions_immd</td>
<td>100%</td>
<td>436 / 436 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_A_D_scan_latch_assertions_immd_prev</td>
<td>100%</td>
<td>83 / 83 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_D_D_ISO_ASSERTIONS</td>
<td>100%</td>
<td>150 / 150 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_D_D_POR_ASSERTIONS</td>
<td>100%</td>
<td>238 / 238 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_HIBERNATE_SEQ_CHECK</td>
<td>100%</td>
<td>36 / 36 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
<tr>
<td>u_POWERUP_SEQ_CHECK</td>
<td>100%</td>
<td>15 / 15 (100%)</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>100%</td>
</tr>
</tbody>
</table>

- 100% coverage on vplan and auto-generated vplan from CPF

<table>
<thead>
<tr>
<th>Name</th>
<th>Overall</th>
<th>Goal</th>
<th>Overall</th>
<th>Planned Elements</th>
<th>Mapped Elements</th>
<th>Mapped Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>pmu.vplan</td>
<td>100%</td>
<td>1</td>
<td>358 / 358 (100%)</td>
<td>112</td>
<td>112 / 112 (100%)</td>
<td>100%</td>
</tr>
<tr>
<td>Reference CPF Auto vplan</td>
<td>100%</td>
<td>1</td>
<td>538 / 538 (100%)</td>
<td>204</td>
<td>204 / 204 (100%)</td>
<td>100%</td>
</tr>
</tbody>
</table>
Results

- Verification closure by 3-4 month to tapeout
- Healthy silicon with zero bugs
Conclusion

- Proposed solution helped us in
  - Expose bugs early
  - Exhaustive verification of isolation (including analog to analog)
  - Exhaustive verification of Analog control sequencing.
  - Coverage closure without Cosims