Experience with OVM-Based Mixed-Signal Verification of the Impedance Calibration Block for a DDR Interface

Harry Wang
Microsemi Corporation
San Jose, California
Harry.Wang@microsemi.com

Wessam El-Naji
Mentor Graphics Corporation
Cairo, Egypt
Wessam_El-Naji@mentor.com

Kenneth Bakalar
Mentor Graphics Corporation
Fremont, California
kenneth_bakalar@mentor.com

***Abstract*— We qualified a structured approach to mixed-signal system-on-chip (SoC) verification using systematic pre-planning and the Open Verification Methodology (OVM). The special requirements of interfacing to the mixed-signal design under test (DUT) are encompassed by a library of driver and monitor extension elements that we call *O-SRC* and *O-PRB*. We report on the verification plan, the development effort, the results achieved, and our conclusions regarding the viability of these techniques for future product development at Microsemi.**

Keywords-OVM; analog mixed-signal; interface; driver and monitor library; verification plan

I. INTRODUCTION

Microsemi Corporation's products include mixed-signal FPGAs that are complex and highly programmable, with many potential operating modes. The devices are embedded by customers in diverse environments limited only by the engineer's imagination.

We report on the qualification of a new methodology for systematic metric-driven verification of these devices. The methodology uses structured preplanning, the OVM augmented by a library of analog sources and probes, and the Questa ADMS simulator from Mentor Graphics Corporation. We aimed to realize reduced total verification effort and improved quality while establishing a foundation of reusable verification components for future projects.

II. THE DESIGN UNDER TEST

For the purposes of this evaluation we chose a mixed-signal block that is included in our new class of mixed-signal FPGAs. The block calibrates the impedance of the IO drivers of an on-chip DDR interface to an external, application-dependent reference network. A state machine is used to identify the codes that cause the output to match the desired impedance. Calibration is performed at power-up and may be repeated later to account for temperature or voltage drift.

The calibration block is illustrated in Fig. 1. A surrogate for the driver on-die termination pull-up resistor network, Z_P, is tied to ground through an external 1 percent tolerance reference resistor. The output of this network is sent to a comparator. The output of the comparator is sent to a digital state machine that uses the comparison result to adjust the Z_P trim bits, *pcode*. The comparator trips when the voltage at *pad* equals the internal reference voltage, at which time we expect the Z_P impedance to be "trimmed". The *pcode* is then locked and sent to the IO driver to duplicate the same impedance as Z_P.

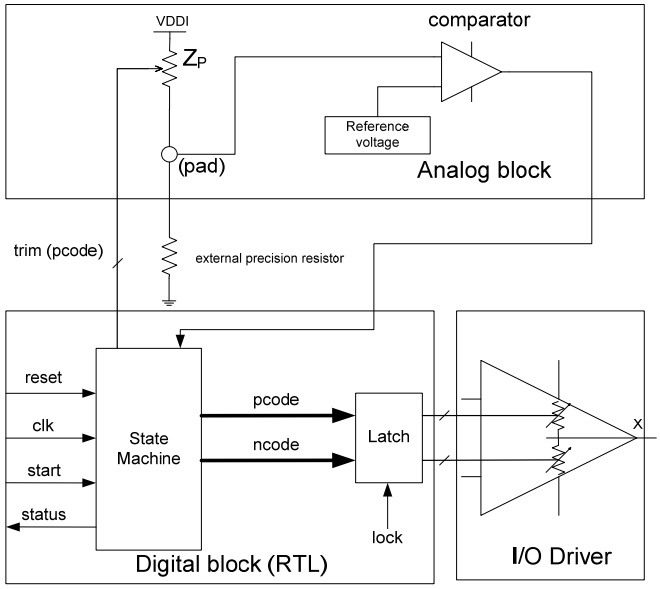

Figure 1. The calibration block.

A similar circuit and process is duplicated for the pull-down network (not shown in the diagram).

III. THE VERIFICATION CHALLENGE

Traditional practice in the verification of mixed-signal DUTs calls for a testbench written using a digital hardware description language (HDL). For analog engineers, this entails mastering the complexities of Verilog or VHDL syntax and the novel notions of function, task, and event driven processes. The analog mixed-signal (AMS) engineer may be overwhelmed when the timing and sequencing of the stimulus becomes complex.

Another challenge lies in gauging the quality and coverage of the mixed-mode simulation. Conventional ways of observ-

ing and processing waveforms is limited, error prone, and hard to reuse.

Finally, the jury-rigged test environment created in this way is very difficult to reuse for testing the next member in the device family. More often than not, the mixed-signal designer starts over from scratch.

IV. PROBLEMS WITH PREVIOUS APPROACH

The original verification of the block was completed using a conventional strategy with an ad-hoc Verilog testbench to drive the process. This proved unsatisfactory for the following reasons.

- The test vectors were coded in-line. This is little better than using SPICE Piecewise Linear (PWL) sources.
- The complexity of the pin-level protocol made it difficult to avoid invalid or conflicting configurations (e.g., enabling both pull-up and pull-down at the same time).
- Automating the checking of results was a challenge. Extra signals were needed to synchronize the result checking with the corresponding stimuli. The asynchronous nature of analog events makes this problem especially puzzling.
- The analog designers struggled with Verilog coding issues. Adding a single new test was a major effort.
- The block level tests devised by analog engineers were difficult to integrate into the top-level testbench.

V. PROPOSED METHODOLOGY

The overall goal of our verification methodology is to enable complete system-level verification for mixed-signal SoCs by integrating the verification of the analog and digital parts of the SoC into a single, mixed-signal verification environment.

There are two requirements for calibrator verification: first, to verify the lock mechanism for the impedance code of the DDR driver block; second, to verify that the lock codes match the desired impedance.

A. The Verification Plan

To ensure that we satisfied all of our verification goals, the first thing we did was adopt a structured approach, which starts with a complete verification plan—a common best-practice in digital design flows. The verification plan encompasses design requirements and verification requirements quantitatively defined for the analog part of the design.

Figure 2. Outline of the verification plan.

Fig. 2 shows the table of contents of the verification plan. The first three sections of the verification plan list the tools used for verification, the libraries used to build the verification environment, and the required compute resources. The fourth section specifies the available product definition documents and human resources.

This is followed in Section 5 by the design requirements. Design requirements are formal, quantitative definitions of the design specifications. They are directly extracted from product documents. Design requirements are prerequisite to the formulation of the successive phases of verification. Our plan calls out 61 distinct design requirements.

Section 6 lists the verification requirements, which are derived from the design requirements. There is a many-to-one relationship between design requirements and verification requirements. Each verification requirement is classified by type (generate, check, or cover) and assigned to the verification infrastructure element that fulfills that requirement. The plan calls out 13 check requirements and four generate requirements, which specify the stimuli that will be needed to fulfill the check requirements.

Fig. 3 illustrates a typical design requirement and the corresponding verification requirements from Sections 5 and 6, respectively (red boxes).

Sections 7 and 8 describe the file organization and the verification infrastructure. These include the environment block diagram, a description of the verification components and their reusability, and the verification layers. The final subsections describe development phases and the schedule for the implementation of the verification environment.

Design Req ID	Description	Reference	Section	Comments Issues
DR_001	Tolerance of external resistance 1%	IO_Calibration. doc	2.1	
DR_002	Reference resistor value should equal to the desired impedance of REFP block	IO_Calibration. doc	2.1	
....				
DR_006	REFN_comparator trips when REFN nmos network imped-ance matches the external resistor	IO_Calibration. doc	2.1	
DR_007	Once PCODE and NCODE trim bits are found they are latched and sent to drivers	IO_Calibration. doc	2.1	
...				

VR_ID	Check	DesReqID	O-SRC	O-PRB	Actors	Method
VR_001	ddr_lock	DR_001 DR_002 DR_006 DR_007 DR_012 DR_013 DR_015 DR_018 DR_022 DR_034 DR_035 DR_036 DR_037 DR_038 DR_039 DR_040 DR_060	DC volt	voltage / current sensing	score-board	Range Checker
VR_002	ddr_no_ lock	DR_058	DC volt	voltage / current sensing	score-board	Range Checker
VR_003	ddr_pc/ dpc	DR_059	None	voltage sensing	score-board	Range Checker
VR_004	ddr_ pow-erdown	DR_060	DC volt	voltage / current sensing	score-board	Range Checker
...						

Figure 3. Design requirements and the derived verification requirements.

B. The Enhanced OVM Environment

We enhanced the time-tested, digital-centric OVM to establish an integrated mixed-signal environment (OVM-A) for system-level verification.

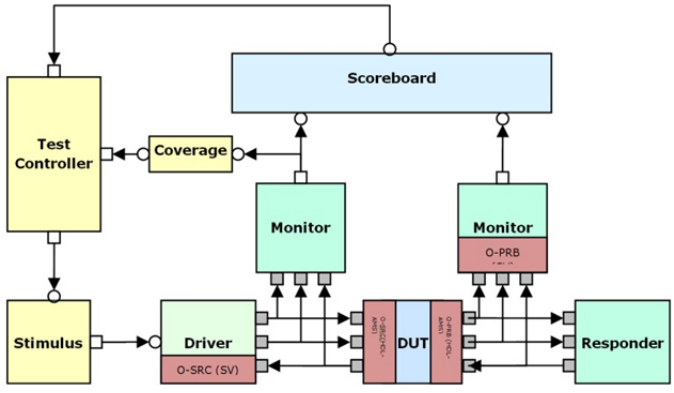

Figure 4. The mixed-signal OVM-A environment using the proposed methodology.

The special requirements of interfacing to the mixed-signal DUT are met by a library of driver and monitor extension elements (O-SRC and O-PRB). The elements are implemented in pairs of a SystemVerilog class and a behavioral AMS module.

The static functions and tasks of the class are used either in the driver or responder (in the case of an O-SRC) or in the monitor (in the case of an O-PRB); the behavioral modules are instantiated in a wrapper encapsulating the instance of the DUT. The two elements communicate with each other across the wrapper interface to supply stimulus or probe results

The O-SRCs (Fig. 5) provide analog stimulus to the DUT. The SystemVerilog class is responsible for interpreting the control parameters to suit the nature of the source. The behavioral module is a signal source or load controlled by digital parameters. In the general case, the SystemVerilog class encapsulating the driver side of the O-SRC may be parameterized.

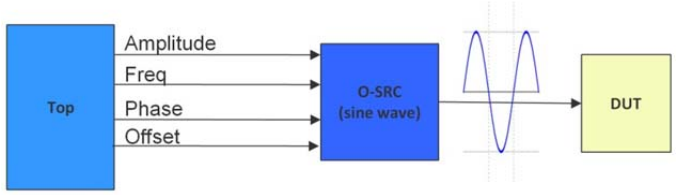

Figure 5. Concept of O-SRC.

An O-PRB (Fig. 6) samples analog output, analyzes the data, and then passes a summary of the results on to the monitor.. Fig. 7a and 7b show the two elements of an O-PRB that measure the output impedance of a selected analog pin. Fig 8 shows the use of that O-PRB in a monitor.

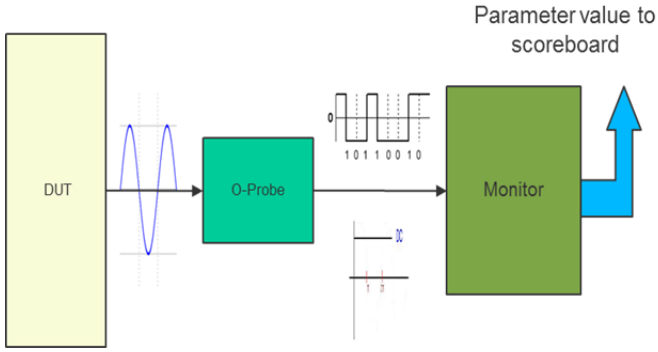

Figure 6: Concept of O-PRB.

The existing library of O-PRBs and O-SRCs, supplied by the simulator vendor and shown in the following bullets, is easily expanded by the verification engineer.

- O-PRB
  - Peak detector
  - Frequency /Rise/Fall time
  - Eye Diagram
  - FFT
  - Phase shift detector
  - Amplitude follower
  - Jitter calculator
  - Current measurement
- O-SRC
  - Voltage and current sources ***(DC, Exponential, Pulse, Voltage Bit Pattern, Piecewise Linear [PWL], Sine wave, Single frequency FM [SFFM], AM)***
  - Voltage Controlled Voltage Source VCVS ***(Linear, Gates, Delay)***
  - Voltage Controlled Current Source ***(Linear)***
  - Current Controlled Voltage Source ***(Linear, Gates, Delay)***

```
module load_curr_oprb (
  curr_measure_out, volt_measure_out,
  probe, hsup, lsup, term_rl
  );
  output curr_measure_out;// I measurement out
  output volt_measure_out;// V measurement out
  voltage curr_measure_out;
  voltage volt_measure_out;
  input probe; // the probe
  input hsup;  // High supply
  input lsup;  // Low supply
  input term_rl;// Ext. termination R value
  electrical probe;
  electrical hsup;
  electrical lsup;
  wreal term_rl;

  electrical half_vdd;
  real  probe_curr;
  branch prb_vdd

  analog begin
    // Generate half VDD supply
    V(half_vdd) <+ 0.5*(V(hsup)+V(lsup));
    // Terminate the probe
    V(probe, half_vdd) <+ term_rl*I(probe,
half_vdd);
    // Extract current
    probe_curr = I(probe, half_vdd);
    V(curr_measure_out) <+ probe_curr;
    V(volt_measure_out) <+ V(probe);
  end

endmodule // load_curr_oprb
```

Figure 7a. O-PRB for load current/voltage measurement, Verilog-AMS part.

```
class load_curr_oprb;

  static function real
      get_imp(real prb, cur, hsup, lsup);
    if ( cur > 0 ) // Pad is sourcing current
      return ( (hsup - prb) / cur );
    else if ( cur < 0 ) // Pad is sinking current
      return ( (lsup - prb) / cur );
    else
      return 1e9;
  endfunction // get_imp

endclass
```

Figure 7b. The same O-PRB, SystemVerilog portion, with a function to calculate the impedance.

The OVM testbench is, as usual, organized in layers (Table 1). The new bottommost layer contains the O-SRC and O-PRB library components associated with driving and sensing the DUT. Above that is a layer of transactors—devices that convert between the transaction-level and pin-level worlds. The components in the layers above the transactor layer are all transaction-level components.

TABLE I. VERIFICATION LAYERS

Layer Name	Layer Tag	Associated Component
Control	L1	Tests
Analysis	L2	Scoreboard
Environment	L3	Environment
Transactors	L4	Agents, Drivers and Monitors
OVM-A Components	L5	O-SRC, O-PRB

VI. A SAMPLE MIXED-SIGNAL TEST

The test we describe here measures the calibrated IO output impedance and compares it with the external resistor connected to the DDRIO calibration block.

A custom OPRB was created to provide a variable resistive load (corresponding to the *external precision resistor* in Fig. 1).

The four blocks of the DDRIO calibrator DUT were enclosed in a Verilog-AMS wrapper with three instances of a digitally controlled O-SRC to provide the necessary power supplies, the resistor O-SRC for the variable load, and two instances of the current probing O-PRB (Figure 7). The outputs of the O-PRBs pass through the ports of the wrapper.

The wrapper itself is instantiated in the top-level testbench, where the probe results are assembled into the transaction that is transmitted to the scoreboard for analysis.

VII. THE TESTBENCH

The testbench follows the normal OVM pattern. Verification components were organized according to design functionality and verification requirements. The four categories of verification components implemented in the extended OVM environment are summarized in Table 2.

TABLE II. MAIN VERIFICATION COMPONENT CATEGORIES

Category	Description
calibration	(agents, transactions, sequences, and monitors) for calibration block.
ddrio	agents, transactions, sequences, and monitors for ddrio block.
share_pd	agents, transactions, sequences, and monitors for shared_pd pins
ovma	agents, transactions, sequences, and monitors for analog stimuli and analog measurements

In addition to the DUT wrapper instance previously described, the testbench included an OVM driver for the real-valued controls of the O-SRCs and a monitor that sends O-PRB results to the scoreboard. As is customary, these were packaged inside a subclass of OVM_agent.

As is indicated in Table II, other agents driving and monitoring other functions of the DUT were also implemented to provide the stable stimulus needed to perform the sample test described here. The same agents will be reused to meet other verification requirements.

VIII. THE MIXED-SIGNAL MONITOR

Recall that the O-PRB module instantiated inside the wrapper generated real values representing the calibrator current and voltage on the interface of the wrapper whenever the calibration impedance changed.

Fig. 8 shows the run task in the monitor. The monitor repeatedly extracts the analog measurements and stimuli and packs them into a transaction at the rising edge of the stimulus clock. The static function of the O-PRB class is called to measure the impedance. Then the assembled transaction is written to an analysis port to make it available to the scoreboard.

```
task run();

  ovma_transaction ovma_item;
  ovma_transaction ovma_item_clone;

ovma_item=ovma_transaction::type_id::create("ovma_it
em");

  // Wait for reset to complete
  @(posedge top_v_if.calib_if.iocalibrst_b);
  // Sample the outputs at clock edge
  forever @(posedge top_v_if.calib_if.clk_50m) begin
  // Wait for calib to complete
   while (top_v_if.calib_if.iocalib_intrpt != 1'b1)
         @(posedge top_v_if.calib_if.clk_50m);
  // Wait one more cycle
   @(posedge top_v_if.calib_if.clk_50m);
  // build the transaction
   ovma_item.x_ext_res_rl =
top_v_if.stim_if.x_ext_res;
   ovma_item.x_vddi_rl = top_v_if.stim_if.x_vddi_rl;
   ovma_item.x_vssi_rl = top_v_if.stim_if.x_vssi_rl;
   ovma_item.padn_voltage =
top_v_if.stim_if.padn_volt_mrl;
   ovma_item.padp_voltage =
top_v_if.stim_if.padp_volt_mrl;
   ovma_item.padn_current =
top_v_if.stim_if.padn_curr_mrl;
   ovma_item.padp_current =
top_v_if.stim_if.padp_curr_mrl;
   ovma_item.padn_imp_rl  =
   load_curr_oprb::get_imp(ovma_item.padn_voltage,
   ovma_item.padn_current,
   ovma_item.x_vddi_rl,
   ovma_item.x_vssi_rl );

   ovma_item.padp_imp_rl  =
   load_curr_oprb::get_imp(ovma_item.padp_voltage,
   ovma_item.padp_current,
   ovma_item.x_vddi_rl,
   ovma_item.x_vssi_rl );

   `ovm_info(get_type_name(),
   $psprintf("OVMA Transfer collected by monitor
:\n%s",
   ovma_item.sprint()), OVM_MEDIUM)

   // Clone the result
   $cast (  ovma_item_clone, ovma_item.clone());
   // Broadcast the cloned item
   ovma_stim_mon_ap.write(ovma_item_clone);

   end //forever
endtask
```

Figure 8. Run task in the monitor instantiating the SystemVerilog part of the O-PRB.

IX. EXECUTION

Testing the impedance measurements entails changing the value of the external resistance, performing calibration twice, and then probing results and judging the results. Five sets of such sequences were executed.

The simulation covers the design requirement DR-002 from table 5-1.

X. THE RESULTS

After fixing a few typical OVM coding bugs, mostly related to constructing and instantiating SystemVerilog objects, we were able to execute the test. We performed the first run with a known defective version of the DUT with an elusive error, actually an early version of the calibrator from the original development effort. As expected, the run produced errors, but our new system made the defect easy to find and diagnose. Fig. 9a shows the ADMS transcript message indicating that there are ten OVM_ERRORs.

```
# --- OVM Report Summary ---
# ** Report counts by severity
# OVM_INFO :   46
# OVM_WARNING :    0
# OVM_ERROR :   10
# OVM_FATAL :    0
```

Figure 9a. Results showing OVM_ERROR in summary.

It was straightforward to extract the first OVM_ERROR from the run transcript, at 16.353 µs (Fig. 9b).

```
# OVM_INFO ../tlm/analysis/ovma_stim_monitor.svh(95)
@ 16353000000:
ovm_test_top.env0.ovma_stim_agnt.ovma_stim_mon
[ovma_stim_monitor] OVMA Transfer collected by
monitor :
#  vssi         0.000000
#  vddi         3.300000
#  ext_res      200.000000
#  padn_current         0.000000
#  padp_current         0.000000
#  padn_voltage         1.650000
#  padp_voltage         1.650000
#  padn_impedance       1000000000.000000
#  padp_impedance       1000000000.000000
# ------------------------------------------
# OVM_ERROR @ 16353000000:
ovm_test_top.env0.scoreboard [Impedance Mismatch]
Ext res is 200.000000 Ohms, but output Imp on N is
1000000000.000000 Ohms
```

Figure 9b. Results showing one OVM_ERROR in detail.

We debugged the DUT using the interactive mode of Questa ADMS around time 16.353 µs. We traced the trimming code and calibration states and eventually found that the polarity of the comparator (see Fig. 1) was reversed from that defined in the specification. When we reran the simulation with the polarity fixed, the results showed zero OVM_ERRORs in the report summary (Fig. 9c). We include an example of the OVM_INFO report of a successful test for comparison with Fig. 9b.

```
# OVM_INFO ../tlm/analysis/ovma_stim_monitor.svh(95)
@ 20033000000:
ovm_test_top.env0.ovma_stim_agnt.ovma_stim_mon
[ovma_stim_monitor] OVMA Transfer collected by moni-
tor :
#  vssi         0.000000
#  vddi         3.300000
#  ext_res      200.000000
#  padn_current         -0.004133
#  padp_current         -0.004133
#  padn_voltage         0.823450
#  padp_voltage         0.823450
#  padn_impedance       199.250003
#  padp_impedance       199.250003
# ------------------------------------------------
# OVM_INFO
../tlm/analysis/ddrio_calib_scoreboard.svh(92) @
20033000000: ovm_test_top.env0.scoreboard [Impedance
calibration succeeded on N side ] External Resistor
is set to 200.000000
# OVM_INFO
../tlm/analysis/ddrio_calib_scoreboard.svh(105) @
20033000000: ovm_test_top.env0.scoreboard [Impedance
calibration succeeded on P side ] External Resistor
is set to 200.000000
…
…
…
# --- OVM Report Summary ---
# ** Report counts by severity
# OVM_INFO :   56
# OVM_WARNING :    0
# OVM_ERROR :    0
# OVM_FATAL :    0
```

Figure 9c. Results with corrected DUT.

XI. EVALUATION

The calibration trial case we picked for our evaluation is, in fact, a relatively simple design. From the single project development point of view, people may not see significant advantages with using OVM-A, since there was some overhead when we were first creating OVM components. However, we will be able to leverage this overhead cost by reusing testbench components and the methodology over multiple projects. We will leverage our experiences from this trial in order to efficiently and effectively apply the OVM-A methodology to larger, more complex mixed-signal designs, such as those with multiple, complex AMS blocks that control the non-volatile memory access. The methodology will help gauge coverage progress and increase the coverage of such mixed-signal blocks by extending the test scenarios. Again, when performing chip-level tests, we can directly reuse most of this setup to achieve the same quality of mixed-signal verification as at the integration level.

A. Comparison to experience with previous verification approach

1) Coverage: The sequences and transactions we defined to impose valid stimuli made it possible to randomize the tests with constraints. This increased coverage by exercising more modes, data, and external reference values.

The error uncovered using this technique was not detected by our original verification approach because the pull-down path was not exercised. We could identify the issue by pulling out the waveform and checking the ncode, but this required displaying the internal signals, which we did not want to do in a production flow.

On the other hand, the OVM environment randomized the data sent to the IO driver, revealing the problem. We found that the OVM produced a higher rate of bug discovery by applying constrained randomization.

We are in the progress of adding a coverage scoreboard to collect coverage metrics. In the mixed-signal setup, with the help of O-SRC and O-PRB, the metric-driven coverage approach is now manageable.

Using the coverage data, we can systematically analyze whether we have covered all the design requirements in Fig 5-1. The coverage information collected will help engineers close the gap between design and verification.

2) *Reusability of the Resulting Testbench for Other Projects:* Since the OVM is a layered environment and the OVM-A library (with the O-SRC and O-PRB) adheres to this layered structure, the testbench is highly reusable. In our case, at during chip-level verification, all the transactions based on ovm_sequence_item, sequences, scoreboard, and coverage_scoreboard will be reused. Tests can be reused as well, if they are not too closely associated with other functional blocks at the chip level. For other designs that are similar but have different characteristics, we expect to make some incremental changes in the driver and monitor, which translate the transactions into levels and vectors. The rest of OVM components would be reused directly without modification.

3) Acceptance of New Flow by Analog Engineers and Management: It is not difficult for engineers without extensive SystemVerilog eperience to make modifications using an existing template, such as the one shown below. Therefore, we have seen growing acceptance by our analog engineers in using SystemVerilog on other projects.

```
start_item(req);
assert(req.randomize());
   req.x_vddi_rl = 3.3;
   req.x_vssi_rl = 0.0;
   req.x_vref_rl = 1.65;
   req.clk_dly_num = 5;// 100ns(50MHz clk)
finish_item(req);
```

Our management highly supports the flow since the testbench is highly reusable and coverage is metric driven. In addition, it makes automated regression easier to achieve.

B. Coding Effort

Coding the OVM environment components was difficult at the beginning. After finishing a few components, the rest of the coding effort was pretty straightforward since a lot of code is similar. The driver, monitor, and scoreboard involved a good deal of SystemVerilog coding in order to correctly process the information being sent and received. It also took some effort to correctly arrange the test sequences and synchronize the sequences as needed. We expect much less effort when we are porting the environment to chip-level verification and for other projects.

Although the schedule to build the first OVM environment was unpredictable for various reasons, including the lack of the required skill set and the adoption of new concepts, the later migration was highly predictable. We expect the migration to the chip level to take only one week.

As we have become more experienced with advanced OVM techniques and the OVM-A library components, we feel confident that we will avoid any problems using this solution on other projects. For example, we know how to create easy-to-use templates (for higher level sequences) to simplify complex mixed-signal sequences.

XII. SUMMARY

A new mixed-signal verification methodology that extends the OVM digital verification methodology has been introduced in this paper. We validated the efficacy and benefits of this enhanced OVM-A methodology using a mixed-signal DDRIO block.

Our evaluation showed that an OVM-A flow using the Questa ADMS mixed-signal simulator moves mixed-signal verification from the qualitative to the quantitative domain, reduces development time, and removes analog circuit details from SoC verification. This should reduce design cycle and cost while increasing verification quality.

XIII. REFERENCES

[1] UVM/OVM Verification. Mentor Graphics Verification Academy. http://verificationacademy.com/course-modules/uvm-ovm-verification

[2] UVM/OVM Online Methodology Cookbook. Mentor Graphics Verification Academy. http://verificationacademy.com/uvm-ovm

[3] SystemVerilog Unified Hardware Design, Specification, and Verification Language. IEC 62530 Edition 2.0 2011-05 IEEE Std 1800 ,