DVCon 2012

Design & Verification Conference & Exhibition

February 28 – March 1, 2012

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

Harry Wang
Microsemi Corp.

Wessam El-Naji
Mentor Graphics

Kenneth Bakalar
Mentor Graphics

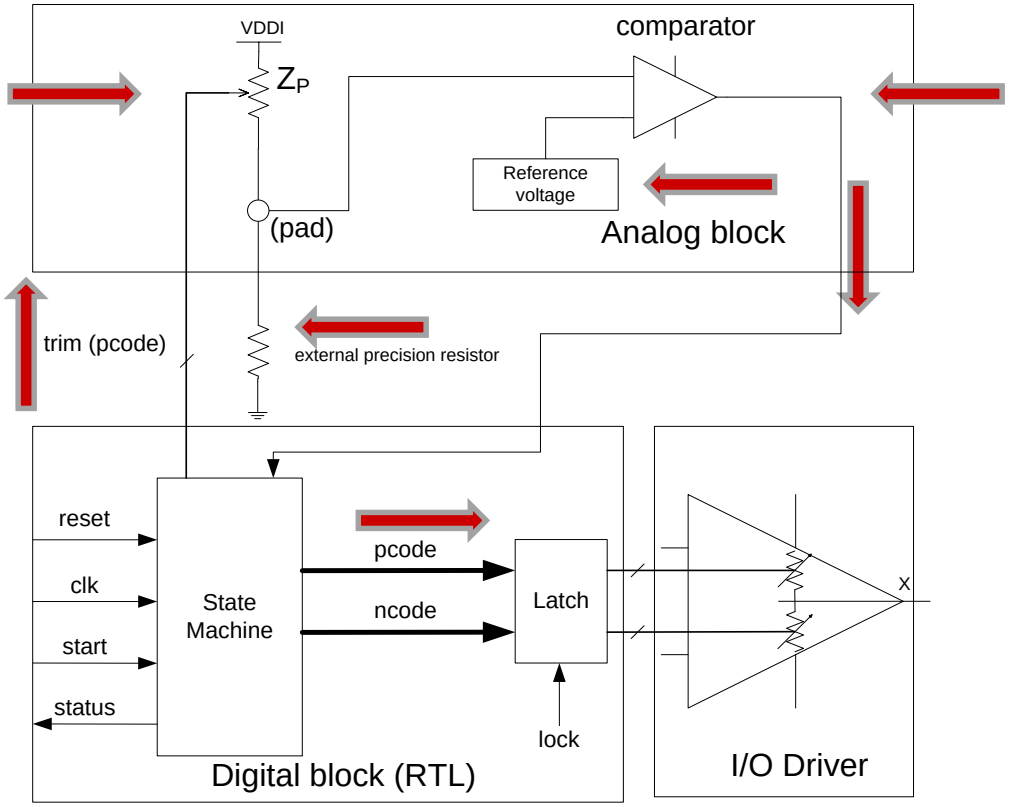
Here's the Story

- Microsemi
- The design under test
- What's wrong with traditional mixed-signal verification
- The new verification methodology
  - Construction of a verification plan
  - Enhancements to OVM
- The results
- What we have learned

The Context

- Microsemi mixed-signal FPGAs
  - Complex
  - Highly programmable
  - Many potential operating modes
- → Complex verification requirements!
- ∴ Explore new strategies for verification
- Start with a small but typical case to prove the concept

The Design Under Test

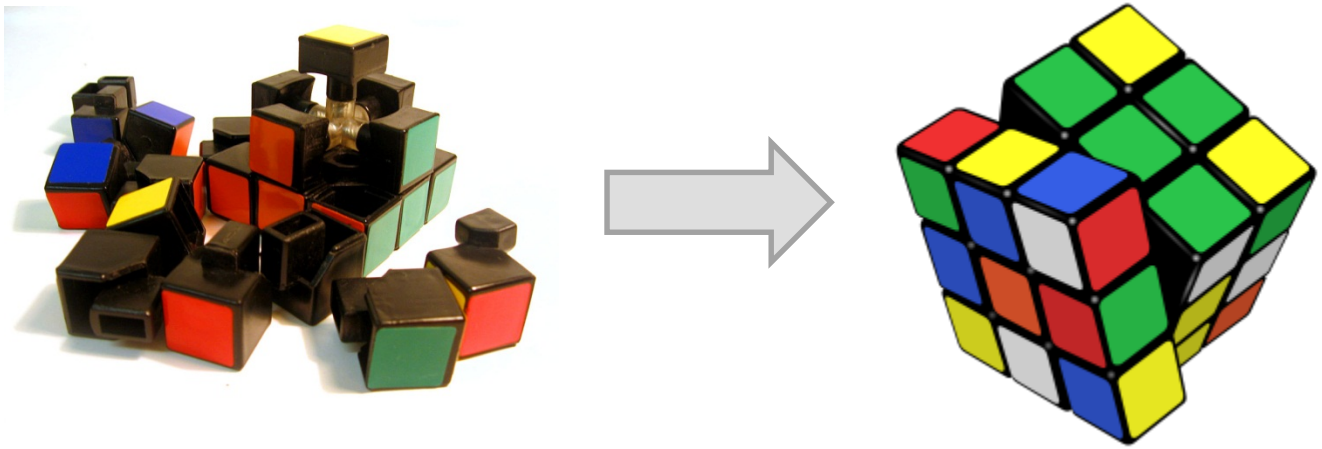


What are the Drawbacks of Tradition?

- Analog engineers must master
  - novel language (VHDL, Verilog, SystemVerilog)
  - novel concepts (process, signal, event)
- Overwhelmed with complex timing and sequencing
- How do I gauge quality and coverage?
- How do I reuse this for the next rev?

Assembling the Puzzle…

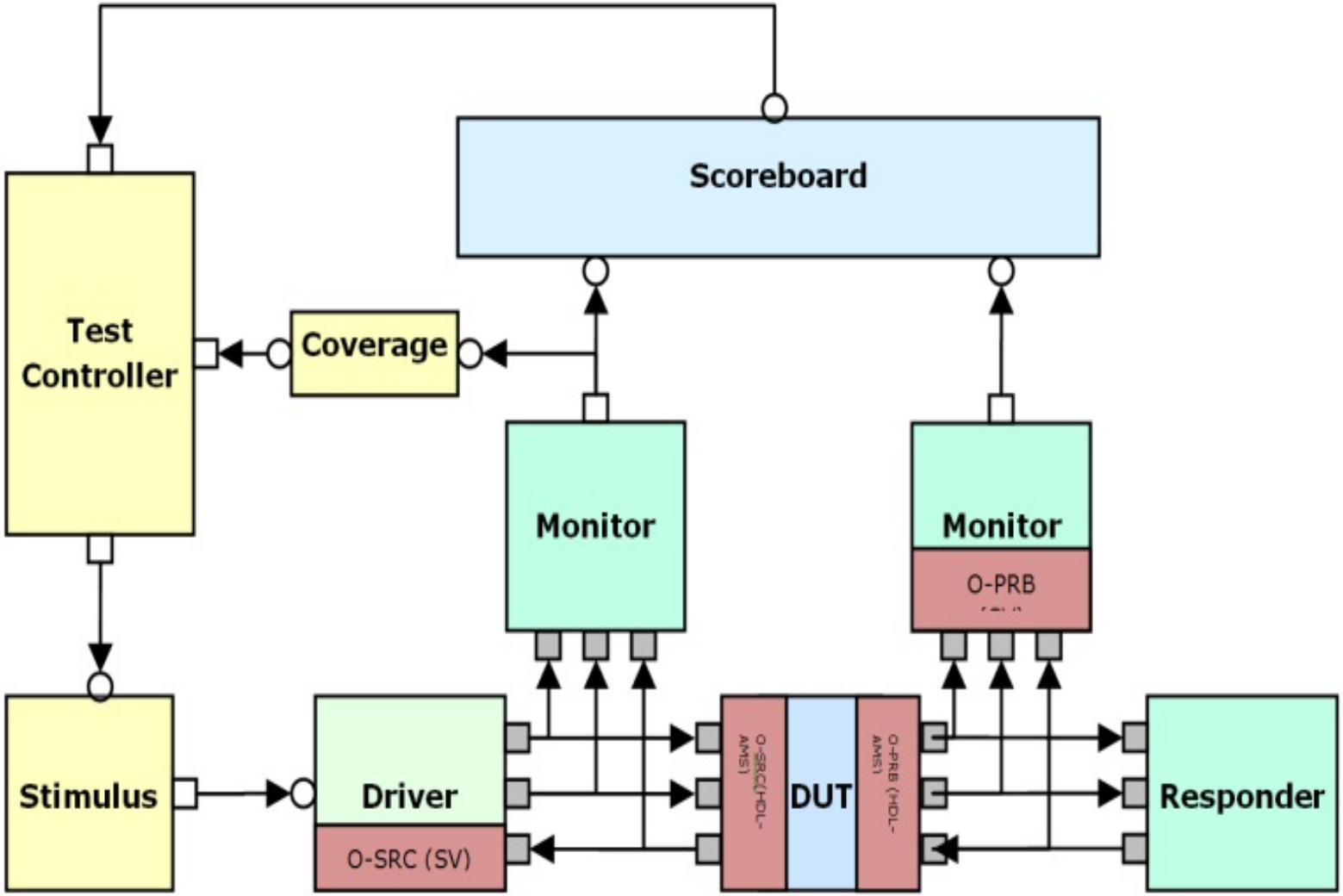
- Test vectors coded in-line
- Complex control protocol
- Synchronizing automated test of analog results
- Big effort to add a single new test
- How do we integrate block level tests?



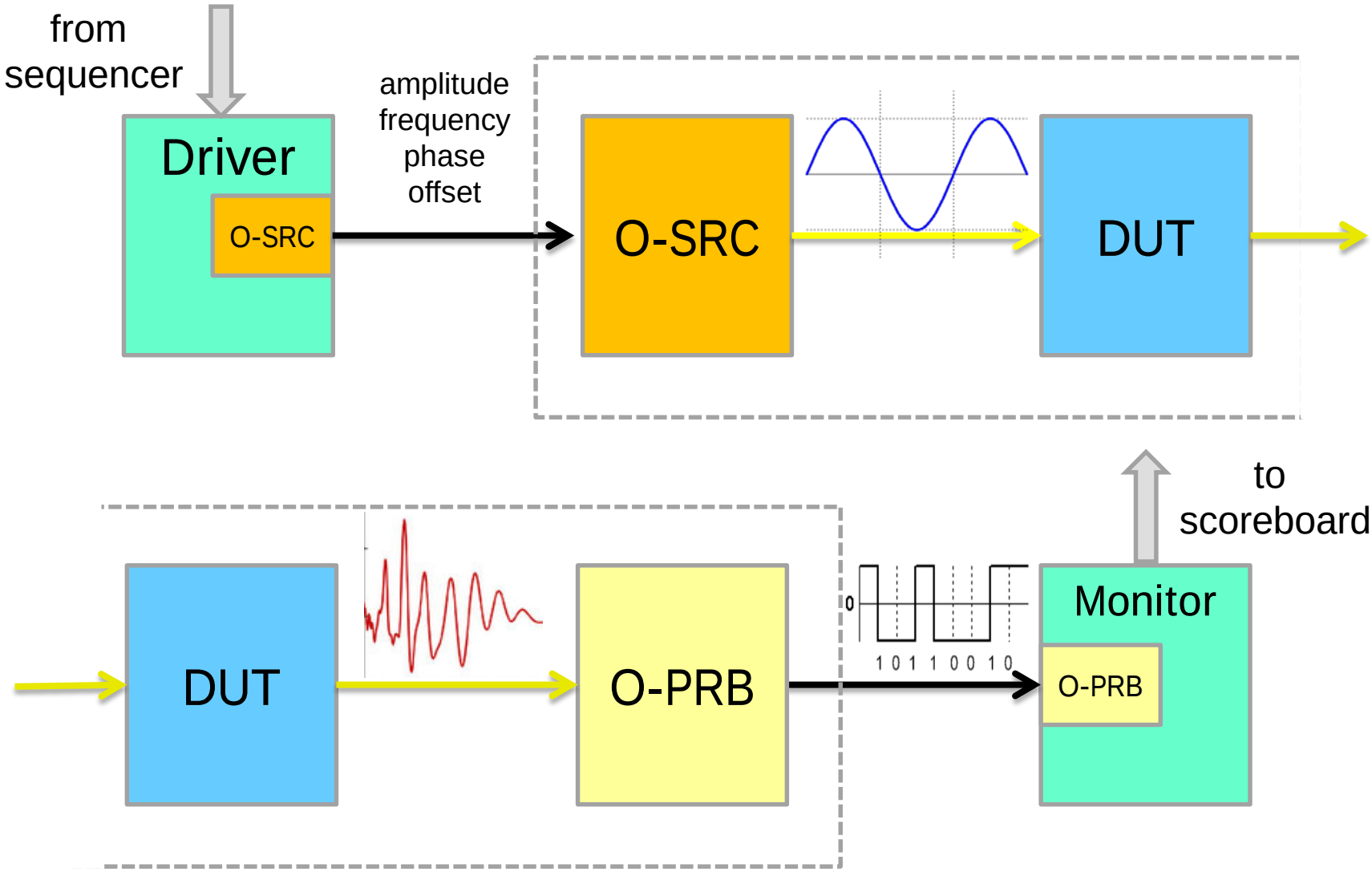
The Verification Plan

1	TOOLS
2	LIBRARIES
3	PLATFORM
4	RESOURCES
4.1	PEOPLE
4.2	DOCUMENTS
4.3	SOFTWARE LICENSES
5	DESIGN REQUIREMENTS
6	VERIFICATION REQUIREMENTS
6.1	CHECKING VERIFICATION REQUIREMENTS
6.2	GENERATION VERIFICATION REQUIREMENTS
7	VERIFICATION INFRASTRUCTURE
7.1	BLOCK DIAGRAM
7.2	REUSE
7.3	VERIFICATION LAYERS
7.4	TABLE OF VERIFICATION COMPONENTS
7.5	DIRECTORY STRUCTURE
8	PHASES AND TIMING
8.1	PHASES
8.2	TASK DISTRIBUTION PROPOSALS

Extensions to the OVM Environment



Sources and Probes



OSRC/OPRB Library

- O-PRBs
  - Peak, gain compression, 3rd order intercept, rise and fall time, frequency, phase shift, jitter...
- O-SRCs
  - DC, exponential, pulse, bit pattern, PWL, sine, FM, AM, VC voltage and current sources, programmable loads...
- User-defined
  - Variants
  - Novel applications

O-PRB (DUT wrapper)

```
module load_curr_oprb (
  curr_measure_out, volt_measure_out,
  probe, hsup, lsup, term_rl
  );

    ...
  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
    V(curr_measure_out) <+ I(probe, half_vdd);
    V(volt_measure_out) <+ V(probe);
  end
endmodule // load_curr_oprb
```

O-PRB (monitor side)

```
class thevenin_equivalent;

  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
```

The Monitor

```
task run();
  ovma_transaction ovma_item;
  ovma_transaction ovma_item_clone;
  ovma_item = ovma_transaction::type_id::create("ovma_item");
  @(posedge top_v_if.calib_if.iocalibrst_b);
  forever @(posedge top_v_if.calib_if.clk_50m) begin
    while (top_v_if.calib_if.iocalib_intrpt != 1'b1)
      @(posedge top_v_if.calib_if.clk_50m);
    @(posedge top_v_if.calib_if.clk_50m);
    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.padn_current = top_v_if.stim_if.padn_curr_mrl;
    ovma_item.padn_imp_rl  = thevenin_equivalent::get_imp(
            ovma_item.padn_voltage,
            ovma_item.padn_current,
            ovma_item.x_vddi_rl,
            ovma_item.x_vssi_rl
            );
    $cast(ovma_item_clone, ovma_item.clone());
    ovma_stim_mon_ap.write(ovma_item_clone);
  end
endtask
```

The DUT Wrapper

```
module ddrio_calib_ams_top(
  // Outputs
  padp_curr_mrl, padn_curr_mrl, padp_volt_mrl, padn_volt_mrl, ...
  // Inouts
  x_padn, x_padp, ...
  // Inputs
  ... , x_ext_res);
  ...
  G4M_DDRIO_CALIB IG4M_DDRIO_CALIB (.clk_50m(clk_50m),...);
  vdc_rl v_x_vref (.vdd(x_vref),.vdd_rl(x_vref_rl));
  vdc_rl v_x_vssi (.vdd(x_vssi),.vdd_rl(x_vssi_rl));
  vdc_rl v_x_vddi (.vdd(x_vddi),.vdd_rl(x_vddi_rl));
  wreal padp_curr_mrl, padn_curr_mrl, padn_volt_mrl, padp_volt_mrl;
  load_curr_oprb padn_curr_oprb (
    .probe(x_padn), .hsup(x_vddi), .lsup(x_vssi), .term_rl(x_ext_res),
    .curr_measure_out(padn_curr_mrl), .volt_measure_out(padn_volt_mrl));
  load_curr_oprb padp_curr_oprb (
    .probe(x_padp), .hsup(x_vddi), .lsup(x_vssi), .term_rl(x_ext_res),
    .curr_measure_out(padp_curr_mrl), .volt_measure_out(padp_volt_mrl));
  // External Precision Resistor
  res_rl x_refp_res(.p(x_refp), .n(x_vssi), .rl(x_ext_res));
endmodule
```

A Sample Sequence

```
class ovma_stim_sequence extends ovm_sequence #(ovma_transaction);
  `ovm_object_utils(ovma_stim_sequence)
  function new (string name="ovma_stim_sequence");
    super.new(name);
  endfunction
  ovma_transaction req;
  …
  task body();
    req = ovma_transaction::type_id::create("req");
    repeat (no_reqs) begin
      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;
      finish_item(req);
    end
      `ovm_info("OVMA Sequence", $sformatf("Stim set to: VDDI =%0f, ...)
  endtask
endclass
```

A Sample OVM_test

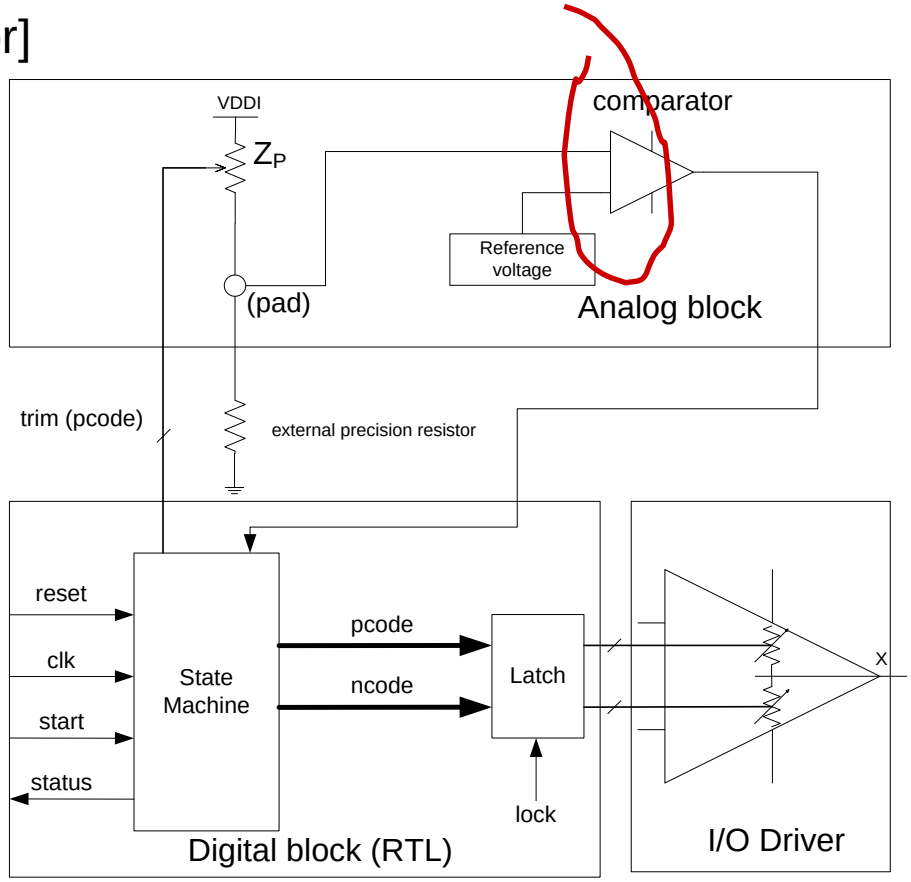
```
class ddrio_calib_test extends ddrio_calib_base_test;
  `ovm_component_utils(ddrio_calib_test)
  calibration_sequence test_seq;
  ovma_stim_sequence   ovma_seq;
  shared_pd_sequence   pd_seq;
  ddrio_sequence       ddrio_seq;
  rand int num_tests=5;
  function new( …
  function void build() …
  task run;
    test_seq  = calibration_sequence::type_id::create("test_seq");
    ovma_seq  = ovma_stim_sequence::type_id::create("ovma_seq");
    pd_seq    = shared_pd_sequence::type_id::create("pd_seq");
    ddrio_seq = ddrio_sequence::type_id::create("ddrio_seq");
    repeat (num_tests) begin
      ovma_seq.start(env0.ovma_stim_agnt.ovma_stim_sequencer);
      pd_seq.start(env0.shrdpd_agnt.shrdpd_sequencer);
      ddrio_seq.start(env0.ddrio_agnt.ddrio_sequencer);
      test_seq.start(env0.calib_agnt.calib_sequencer);
    end
    #1000
    global_stop_request();
  endtask
endclass
```

Errors!

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

Nailing the Error

```
# OVM_INFO …@ 16353000000:
….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:
…scoreboard [Impedance Mismatch]
Ext res is 200.000000 Ohms,
but output Imp on N is
1000000000.000000 Ohms
```



Success

```
# OVM_INFO …OVMA Transfer collected by monitor :
#  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 Report Summary ---
# ** Report counts by severity
# OVM_INFO :   56
# OVM_WARNING :    0
# OVM_ERROR :    0
# OVM_FATAL :    0
```

Evaluation by Microsemi

- The calibration trial case is a relatively simple design
  - Costly overhead in first creating OVM components
  - No immediate, significant advantages
- We will be able to leverage this overhead cost for larger, more complex mixed-signal designs
  - Reusing testbench components
  - Reusing methodology
  - Reusing block-level tests at system level
  - Simple to add test scenarios
- Additional advantages will accrue as we become more sophisticated
  - Coverage metrics
  - Verification progress metrics