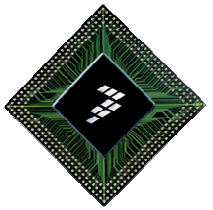
cdn LIVE EMEA 2010

Mixed-Signal Verification Challenges

CDN Live 2010 - Munich



Regis SANTONJA

freescale semiconductor

Presentation

- I am Regis Santonja
  - Mixed-signal Verification Leader since 2003
  - Previously Digital Design Engineer in mixed-signal Power-Managerment ICs @ Freescale/Motorola since 1998
  - Previously Digital Design Lead @ CSTI
  - Previously Digital Design Engineer @ LSI LOGIC
  - I started as an Applications Engineer @ LSI LOGIC in 1994
  - 16 years of experience in the semiconductor industry

- Working at Freescale, in the Consumer Sensor division
  - Accelerometers
  - Pressure
  - Etc…

freescale™ semiconductor

Agenda

- The IC
- The Test bench
- A progressive Verification Approach: from *wreal* to full transistor
- Merging simulation environments with eManager
- Introducing random into mixed-signal simulations
- Simulation Coverage for analog and digital
- Mixed-level Verification challenge:
  - bringing as much of digital advanced verification techniques to the mixed-signal world
  - some thoughts about closed-loop coverage-driven mixed-signal simulations
- Conclusion

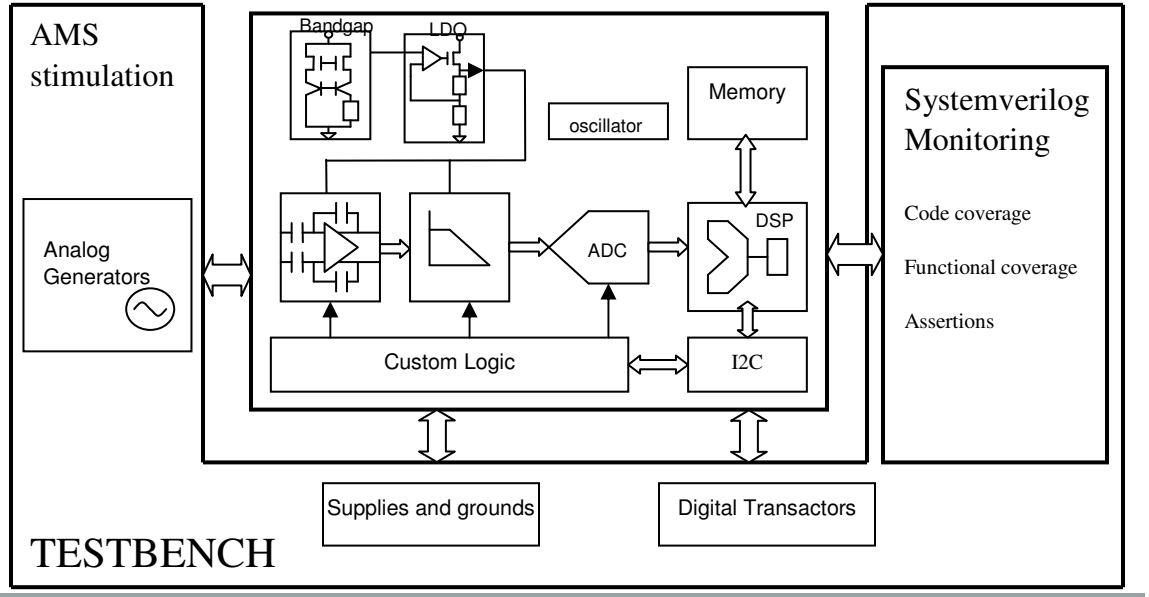
The chip

- Low-power three axis accelerometer for consumer applications
  - Mobile phones
  - Gaming
  - High end pedometer
- Programmable G scales
  - +/- 2g, +/- 4g, +/- 8g
- Multiple Output Data Rate (ODR)
  - From 1.5Hz (low current) to 800Hz
- FIFO buffer
- Automatic embedded detection:
  - Orientation
  - Motion
  - Free-Fall
  - Single and Double Taps
  - Flick, shake, Tilt control, complex gesturing

freescale™ semiconductor

The testbench

- Hierarchical testbench
  - Verilog-AMS top-level
  - Systemverilog sub-block
- Stimulus File
  - Each specific test is coded in a stimulus file included in the testbench via a compiler's directive.
  - Test sequence driven by *initial* digital process
  - Analog (including supplies) controlled by digital via *real* values
- External components can be:
  - in Verilog-A/MS (parameters controlled from the stimulus)
  - Spectre or other provider's format
- Monitoring
  - Verilog-AMS
  - Systemverilog



freescale™ semiconductor

A progressive Verification approach

- *Wreal* models of the analog
- Behavioral digital or RTL

A mix between:

- *Wreal*
- AMS models
- Transistors
- RTL

- All analog at transistor-level
- Digital
  - RTL
  - Gate-level
  - Transistor-level

freescale™ semiconductor

Progressive Verification Approach

Development time

1. Digital only
   - Analog blocks modeled in Real Value (*wreal*)
   - Digital speed
   - System Design and architectural exploration
   - Long transient time issues can be caught
   - Drawbacks:
     - No access to operators such as derivative, integration, Laplace, *last_crossing*
     - Signal-flow model only
     - No easy way to model basic components such as resistors or capacitors
2. Mixed-signal
   - As the design progresses and more accuracy is required
   - Starting swapping *wreal* models to AMS or transistor-level
     - low-freq, close to DC blocks (biasing,…)
   - Difficulty with currents between *wreal* and analog domains
     - Need some specific connect modules to represent current sources
     - Cadence R2E_current and symmetrical E2R_current do not respect current sign convention, as they pass the current values with no sign change
3. Full-transistor
   - Track cross-conductions (leakages) between supplies
   - Instabilities, oscillations
   - Floating nodes

freescale™ semiconductor

Traditional simulation environments

- Analog designers
  - Typically run spice/spectre simulations within Cadence Analog Artist
  - Low usage of automatic checks
  - Low usage of regression
  - No coverage monitoring
- Digital designers
  - Typically run simulations from the command line
  - Self-checking
  - Regression testing
  - Code coverage
- Mixed-signal engineers
  - Typically somewhere in between analog and digital habits

A single simulation environment – 1 -

- Cadence eManager can be used as a single front-end
  - Need ability to launch all sorts of simulations from the command-line
  - Self-checking analog tests make them appear in the global Test Coverage metrics
  - All commands gathered in a VSIF (Verification Session Input File) file to eManager

- Digital simulations launched from *make* command with a *Makefile:*

```
ncvlog -file rtl.caf \
       -define stim_file=\"$(TEST).v\" \
       $(COMP_ARGS)
ncelab -file ncelab.caf \
       -coverage a \
ncsim -status \
      -file ncsim.caf \
      -logfile $(TEST).log
      -covoverwrite \
      -covdesign digital \
      -covtest $(TEST) \
      $(SIM_ARGS)
```

A single simulation environment – 2 -

- Mixed-signal, mixed-level simulations launched from a custom script
  - Pre-compiles systemverilog and custom connect modules
  - Builds the appropriate *amsdesigner* command:

```
start_test.pl -lib top_verification \
              -cell top_testbench \
              -view $ATTR(my_config) \
              -test $RUN_ENV(BRUN_TOP_FILES) \
              -run_dir `pwd` \
              -cds_globals generic.cds_globals \
              -define SEED=`perl -e 'print int(rand(1e6))'` \
              -profile
```

  - The *amsdesigner* command is able to read Cadence DFII configuration views
    - Tells which cell view (wreal, RTL, verilog-A/MS, schematic) is to be considered for each cell in each specific simulation
- Analog simulations launched from Freescale internal tool
  - Complex simulation scripts built from visual components in an intuitive GUI
  - Advanced set of automatic measures
  - Automatic comparison versus spec
  - Advanced reporting
  - Command-line ability

Introducing random into mixed-signal simulations

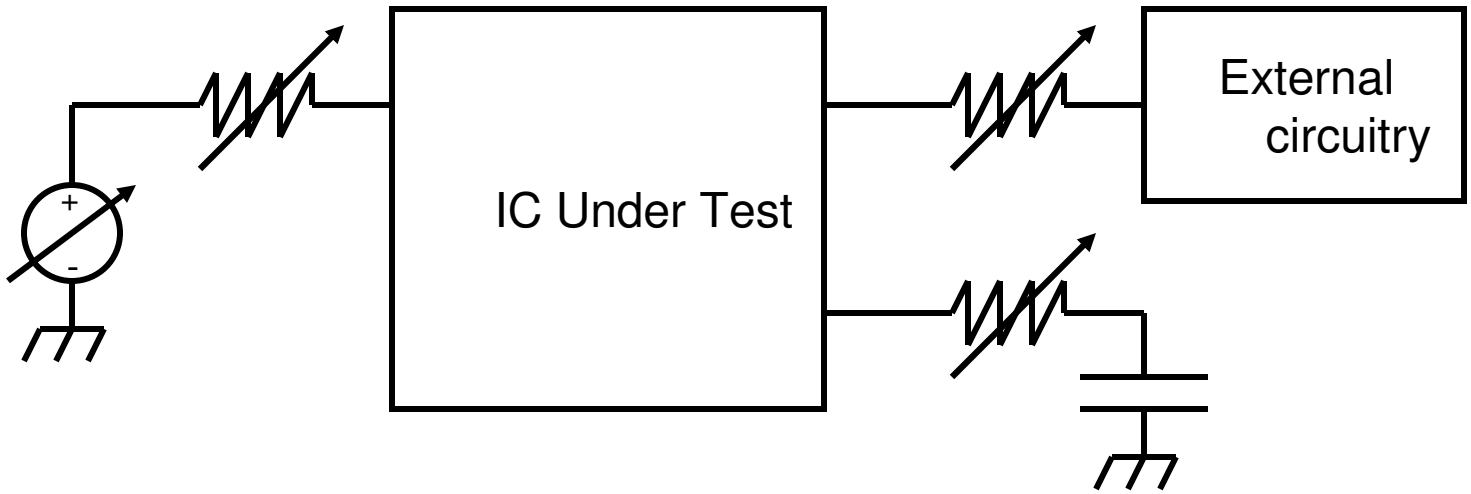
- A *seed* is generated at simulation launch
  - It is recorded in the simulation log file so that we can reproduce a simulation which would show an IC issue
  - It is used to drive, for example, ramp-up/down times of the IC's external power supplies in order to capture potential cross-conduction leakages between supplies

```
initial begin
  VDD_level =0.0;
  `ifdef SEED
    VDD_rise = 100u + 1u*($random(seed) % 100);
    VDD_fall = 100u + 1u*($random(seed) % 100);
  `endif // SEED
  @(event)
  VDD_level = 2.5;
end

analog begin
  V(VDD_stim)      <+ transition(VDD_level, 0 VDD_rise, VDD_fall);
  I(VDD, VDD_stim) <+ V I(VDD, VDD_stim)/RVDD;
end
```

Controlling the analog inputs from the digital

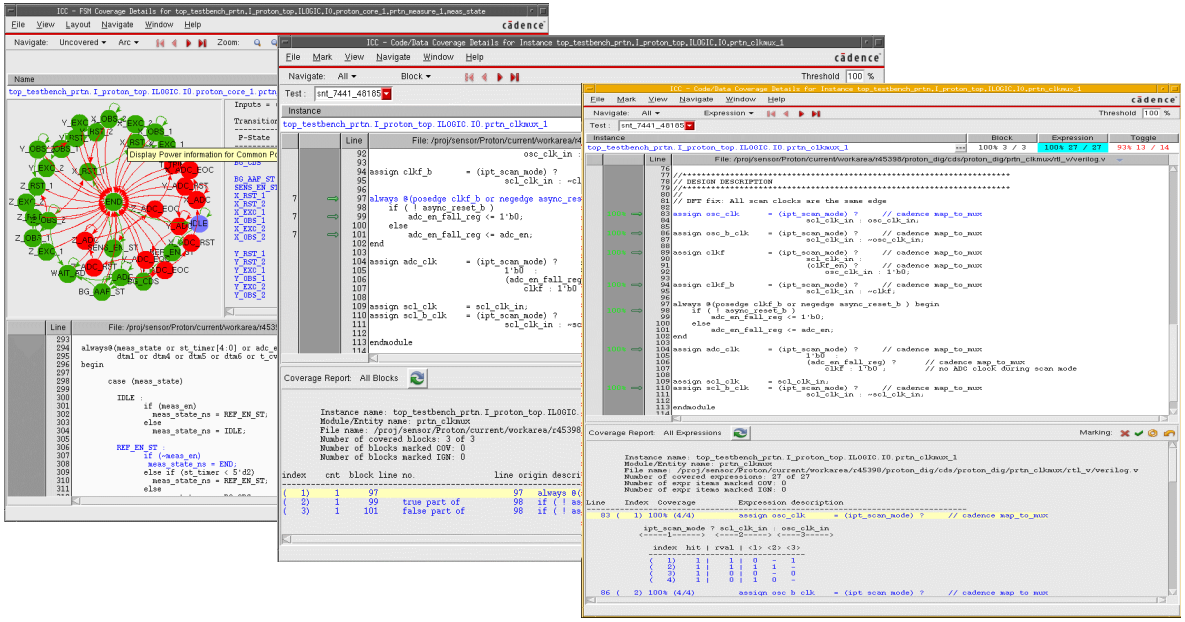
- All analog inputs connected via digital-controlled resistors
- High-valued resistors emulates disconnection
- Resistor value, voltage level, rise and fall times controlled from the digital test sequence (via *initial* process):
- This mechanism allows dynamic connection of more complex external circuitry (i.e. RC filters)

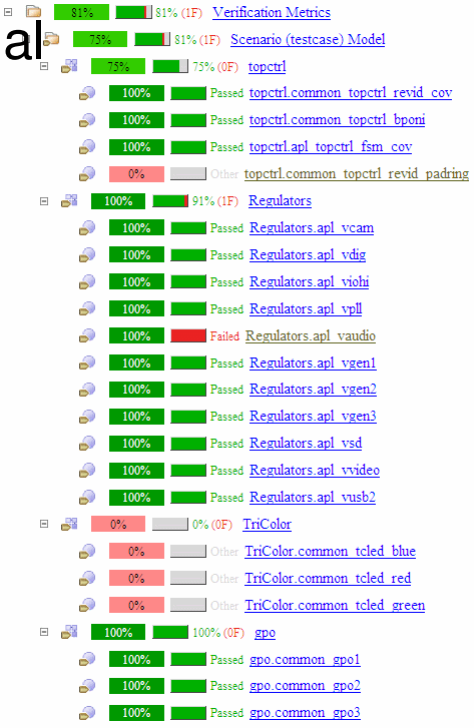


freescale™
semiconductor

Simulation Coverage

- Test Coverage just tells which simulation passes or fails but does not (directly) tell what is verified
- There are two other types of coverage metrics:
  - Code Coverage for digital section
  - Functional Coverage for both digital and mixed-signal





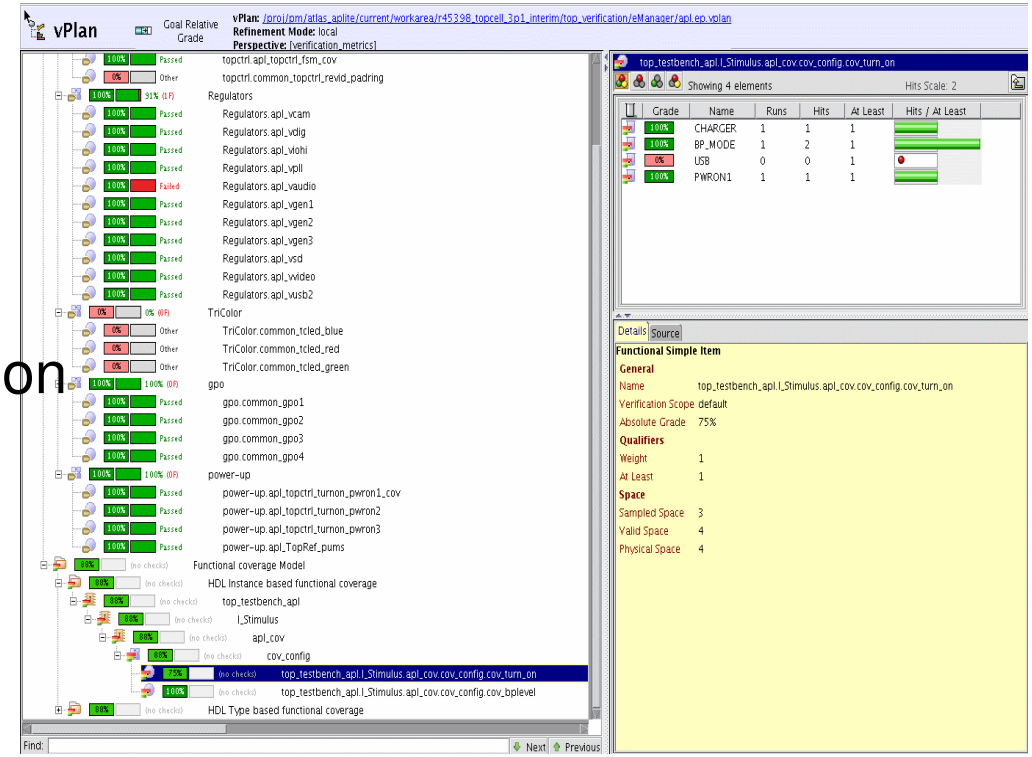
freescale™ semiconductor

Code Coverage

- Code Coverage
  - Automatic
  - Checks which part of the digital code got executed:
    - block, expression, toggle, FSM states and transitions
  - Used ICC (Incisive Comprehensive Coverage Report) to identify holes
  - We could increase our coverage
    - Some functions were never disabled
    - Lots of FSM arcs to the Idle state were missing
    - Lesson learnt: not only is important what the IC should do, but also what it should not do (when the function is disabled or interrupted).

Functional Coverage

- NOT automatic, in opposition to Code Coverage
- Needs to be specified/coded by the user
  - More effort up-front
- It does not depend on the implementation, but on the functionality
- It is closely linked with the specification and the verification plan
- Systemverilog provides constructs to specify and monitor the Functional Coverage



freescale™ semiconductor

Control-oriented Functional Coverage

- The IC measures its acceleration on the 3 dimensional axes
- They are multiplexed in time in order to save hardware
- The control of the signal chain:
  - is a complex time sequence which directly impacts the final measure accuracy
  - Needs to be carefully verified
    - ⇨ Systemverilog concurrent assertion constructs

```
prop_vreg_en : assert property
  (@(posedge clk) $rose(hf_en) |=> ( vreg_en[*72]  ##1
                                    ~vreg_en[*240] ##1
                                     vreg_en[*172] ##1
                                    ~vreg_en[*140])[1:$])
      PRINT_PASS("VREG enable is OK");
  else begin
      $warning("%m fails");
      PRINT_ERROR("");
  end
```

Data-oriented Functional Coverage

- Data-oriented Functional coverage tracks the number of times a design variable reaches a specified set of values
- The design variables to track are called *coverpoints*
- They are gathered into *covergroups*
  - All *coverpoints* in a covergroup are evaluated on the same clock
  - A clock can be:
    - any expression that becomes true
    - The start or end of execution of a named block, a task, a function,…

Data Oriented Functional Coverage – example 1

- We tracked the coverage of our I2c register map
- The actual systemverilog code was automatically generated from an Excel sheet containing the I2c map.
- Here is systemverilog principle:

```
`define F_CNT  1
`define F_MODE 2
`define F_OVF  3

event i2c_write_event;
typedef enum { F_CNT  = `F_CNT,
               F_MODE = `F_MODE,
               F_OVF  = `F_OVF } i2c_field_type;
i2c_field_type i2c_write_field;

covergroup cg_i2c_write @i2c_write_event;
  cov_i2c_write: coverpoint i2c_write_field {option.auto_bin_max = 512;}
endgroup: cg_i2c_write

always @(`STIM.f_cnt) //`STIM refers to AMS testbench top-level
if ($time >0) begin
  @(`DIGITAL_BLOCK.i2c.i2c_write);
  i2c_write_field = `F_CNT;
  -> i2c_write_event;
end
```

Data Oriented Functional Coverage – example 2

- We could also track analog events such as:
  - The internal supply voltage levels
  - The internal voltage references
  - Here is the principle of the systemverilog part:

```
`define VREFON  1
`define VREFOFF 2

typedef enum { VREFON  = `VREFON,
               VREFOFF = `VREFOFF } vref_state_type;
event vref_event;
vref_state_type vref = `VREFOFF;
covergroup cg_vref @vref_event;
  cov_vref: coverpoint vref;
endgroup: cg_vref
cg_vref I_cg_vref = new();
```

  - Here is the principle of the verilog-AMS code that triggers the systemverilog:

```
always @(cross(V(`IC.VREF_adc)-0.9, 0 , 1n, 10u))
begin
  if (V(`IC.VREF_adc) > 0.9)
    covmon.vref = `VREFON;
  else
    covmon.vref = `VREFOFF;
  -> covmon.vref_event;
end
```

freescale™ semiconductor

Mixed-Signal Verification Challenge

- *Wreal* modeling is a means to handle analog-like signals at digital speed
- *Wreal* modeling can be mixed with analog models and/or transistors
- Systemverilog can be used with verilog-AMS to track mixed-signal Functional coverage
- Systemverilog enables dynamic coverage monitoring with dedicated tasks, such as *$get_coverage()*

The door of mixed-signal simulations are now open to explore some advanced digital verification tools and methodologies

- Closed-loop, coverage-driven mixed-signal simmulations
  - The quality of the verification is only as good as the quality of the tests
  - Reaching 90% of coverage with directed tests is a big effort
  - Time-to-market is continuously decreasing
    - ⇨ Automatic mixed-signal test generation, based on dynamic coverage monitoring is possible

Conclusion

- We presented how we can verify a mixed-signal IC:
  - With a progressive approach, starting with pure *wreal* digital simulations, then mixing digital and analog, and finally run full-transistor simulations to track leakages and floating nodes.
  - With eManager as a single front-end to launch our complete set of tests (analog, digital and mixed-signal)
  - By dynamically controlling the analog section and the external circuitry from the digital side
  - By introducing random into the analog domain
  - By taking advantage of ICC to analyze the digital code coverage
  - By using Systemverilog constructs to monitor control-oriented and data-oriented Functional Coverage, including analog events monitoring
- Finally we presented some thoughts about developing a closed-loop, mixed-signal coverage-driven flow, as a complement to today's directed mixed-signal tests

freescale™ semiconductor

Special thanks

- I'd like to thank

  - Jean-Claude Mboli from Freescale, who initiated the adoption of the *wreal* modeling in our Sensor Division at Freescale
  - Thierry Nouguier, our AMS-Designer CAD expert at Freescale
  - Patrick Oury from Cadence, for his dedication to supporting our team on systemverilog and eManager