2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Scalable, Re-usable UVM DMS AMS based Verification Methodology for Mixed-Signal SOCs

Nilesh Sonara, Noorulla Mohammad, Poonam Singh

David Stoops, Joseph Fernando and Kartik Sudarshana

Broadcom Limited

BROADCOM

accellera SYSTEMS INITIATIVE

2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Agenda

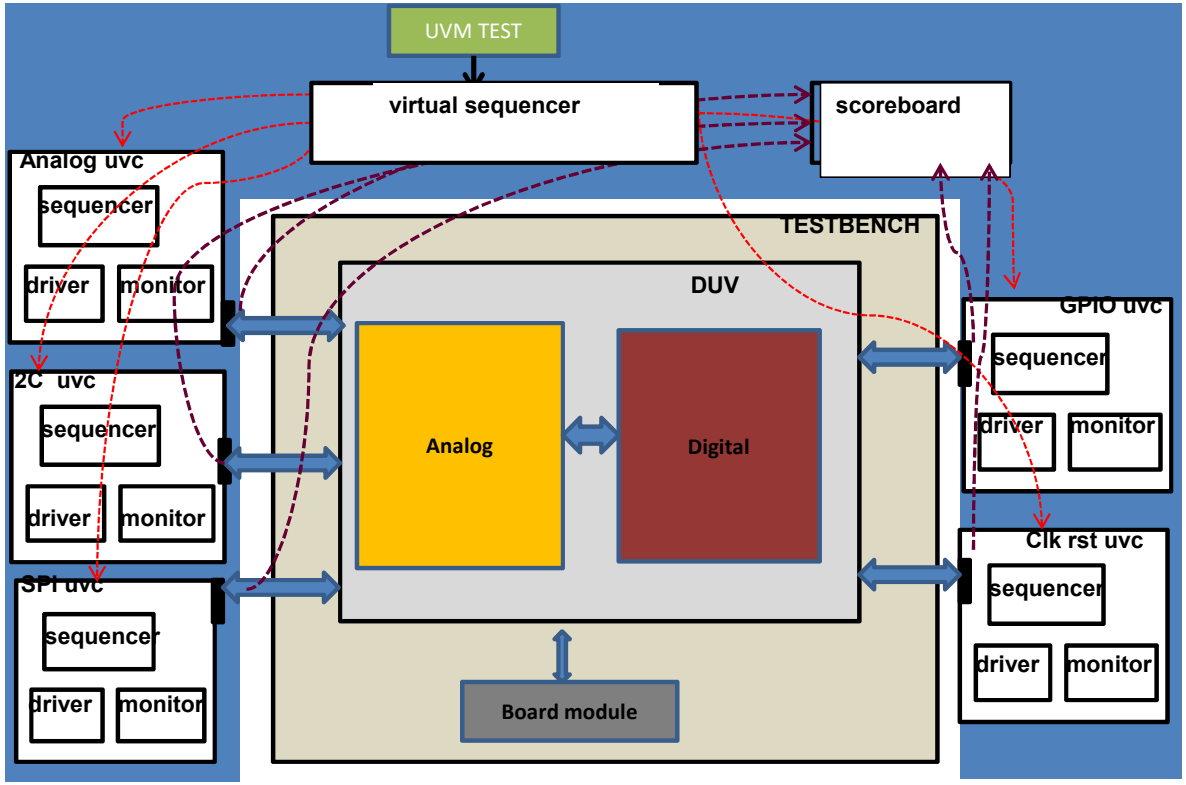
- Motivation
- UVM DMS AMS bench - Reusability
- TestBench Configurations (real, wreal, VerilogAMS, Spice) -Scalability
- Regressions, Simulation data
- Conclusion

accellera SYSTEMS INITIATIVE

Motivation

- Growing complexities in mixed-signal SoCs with more digital and embedded SW based control having multiple feedback loop and interactions with analog
- Simple analog behavior model limit verification and coverage, need for realistic behavior models to cover more scenarios
- Adhoc verification by analog team, need for integrating latest digital verification methodologies for mixed-signal with coverage and checkers.

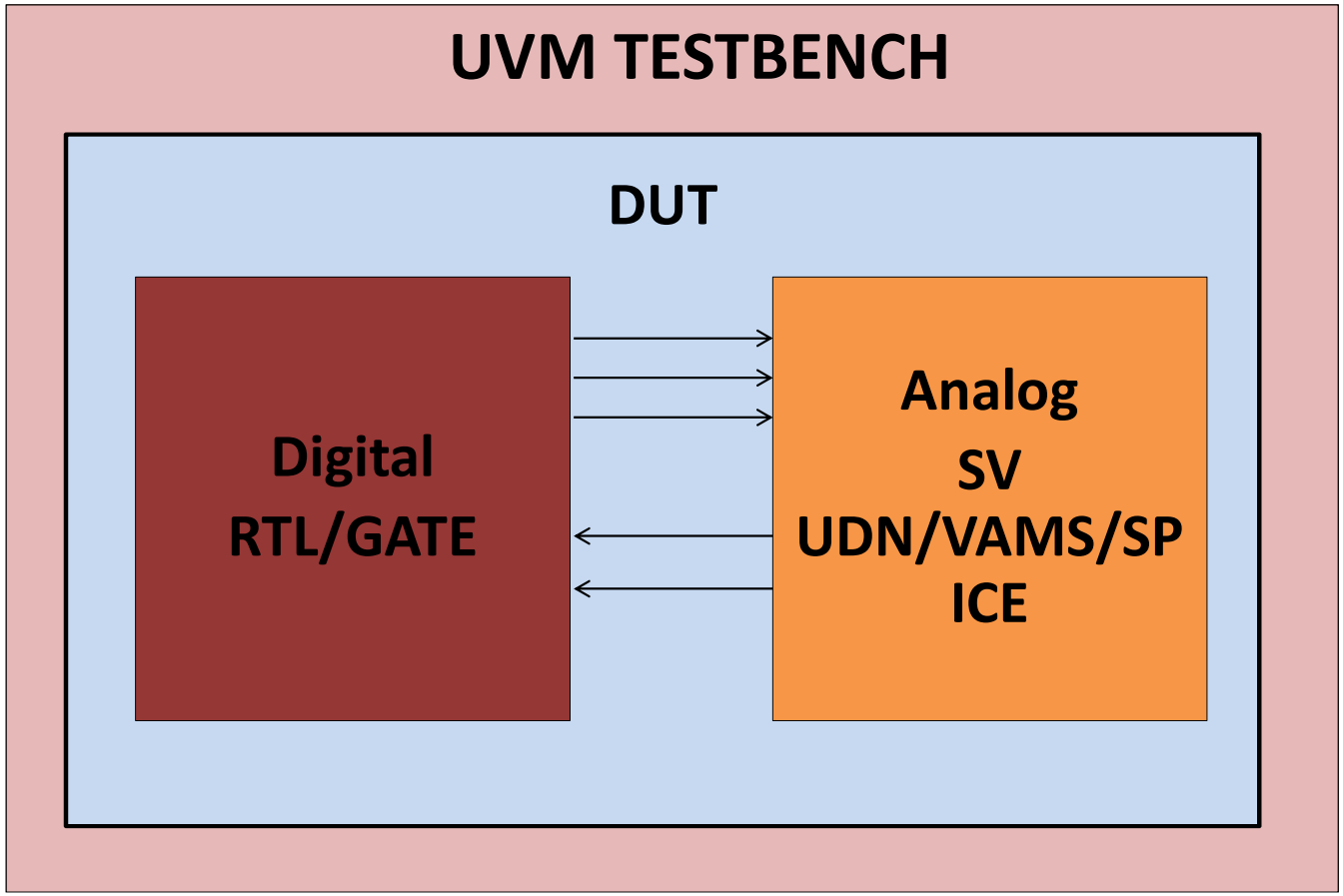
UVM DMS AMS bench



UVM DMS AMS bench

- UVM verification environment is reused for DMS (wreal)and AMS (VerilogAMS and spice) simulations
- Configurable analog UVC to drive and monitor interface activity, example waveform generation with freq, amplitude and phase controls
- Board components
  - Implemented using wreal, re-used for both DMS and AMS
  - Config view replacement is also possible

Testbench Configurations



Testbench Configurations

- UVM Verification testbench reused for different analog model abstraction
- Systemverilog user defined net(SV UDN) based wreal model, VerilogAMS, SPICE and mix of above can be selected
- Same test can be run for any of above configuration.
- Run script takes care of selecting the correct model depending on the argument supplied

Testbench Configurations

- **SV UDN DMS model**
  - allows multi-value nets and multiple drivers.
  - User defined resolution function used to combine multiple outputs together
  - Simulator automatically inserts virtual elements for logic2real and real2logic conversion
  - digital engine without need for analog solver

```
typedef struct{
  real field1; field2; field3;
} T
 function automatic T Tsum(input T driver[]);
 Tsum.field1 = 0.0;
 foreach (driver[i])
   Tsum.field1 += driver[i].field1;
 endfunction
nettype T wTsum with Tsum;
```

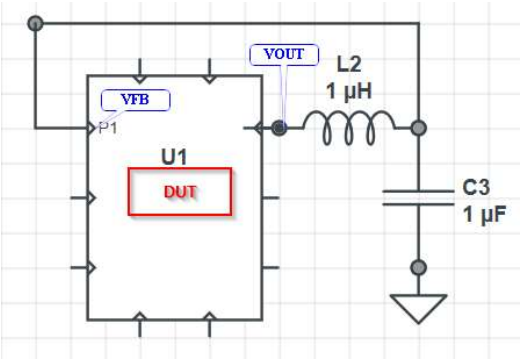
```
Ex: AMS Control File (ie)
amsd{
 ie vsup=1.8 rhi=0.1 rlo=0.1 rout=0.1
 ie vsup=1.1 instport="`top.vdd1p1"
}
```

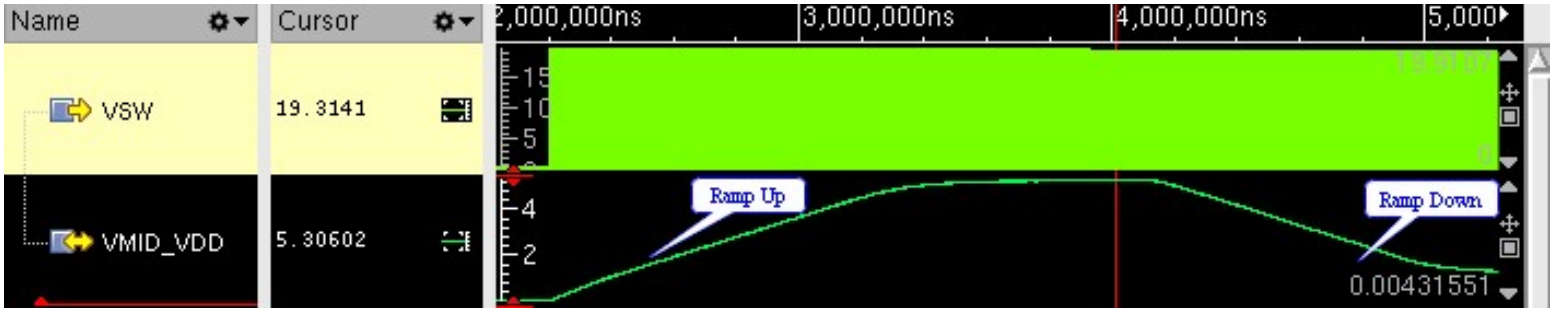
Testbench Configurations

- SV UDN DMS model
  - Bottom up generation from schematic
  - Has supply and ground nets
  - High performance with realistic behavior
  - Used for all functional verification

Testbench Configurations

- SV UDN DMS model : feedback loop example
  - Use of LCR filter to get clean ramping voltages
  - Analog assertions to check slew and voltage ramps
  - Overflow and underflow check
  - Voltage clipping check
  - A2D Assertions
  - Creation of Equivalent C Model for the digital controller path





Testbench Configurations

- VerilogAMS model
  - Replace SV UDN model with verilogAMS
  - UVM tests reused without modification
  - Bottom up generation from schematic
  - Low performance with more accuracy
  - Needs analog solver with digital event based simulator

Testbench Configurations

- Spice model
  - Replace SV UDN model with spice
  - UVM tests reused without modification
  - Bottom up generation from schematic
  - Very slow with very accurate
  - Needs analog solver with digital event based simulator
  - Assertions for checking internal analog nodes

```
// amsd portmap module simulator lang=spectre
amsd { //Default/Fallback discipline
 ie vsup=(1.5) rhi=(0.1) rlo=(0.1) rin=(20M) rout=(0.1) vdelta=(1.5/64.0)
 ie vsup=(1.8) rhi=(0.1) rlo=(0.1) rin=(20M) instport="sys.I_DUT.VDDO"
   portmap subckt=ANATOP porttype=name config cell=ANATOP use=spice
}
```

Testbench Configurations

- Mixed model
  - Selectively choose different model for different modules
  - Use one instance with spice rest with VAMS in case of multiple instance
  - Test decides different configurations

```
simulator lang=spectre
 amsd {       //Default/Fallback discipline
 ie vsup=(1.5) rhi=(0.1) rlo=(0.1) rin=(20M) rout=(0.1) vdelta=(1.5/64.0)
 ie vsup=(1.8) rhi=(0.1) rlo=(0.1) rin=(20M) instport="sys.I_DUT.VDDO"

      portmap subckt=ANATOP porttype=name         config cell=ANATOP use=spice
      portmap module=USB_CPD_top porttype=name         config cell=USB_CPD_top use=hdl
     portmap module=ADC12_top porttype=name         config cell=ADC12_top use=hdl
}
```

Testbench Configurations

- Configurations are created as per test scenario
- CFG2 targets TX module for SPICE simulation

Config	TX	Bias	RX	PMU	MISC
CFG1	VAMS	VAMS	VAMS	VAMS	VAMS
CFG2	SPICE	VAMS	VAMS	VAMS	VAMS
CFG3	VAMS	VAMS	SPICE	VAMS	VAMS
CFG4	VAMS	VAMS	VAMS	SPICE	VAMS
CFG5	VAMS	SPICE	VAMS	SPICE	VAMS
CFG7	SPICE	SPICE	SPICE	SPICE	SPICE

Assertions

- Mixed signal Analog Assertions..
  - Define precision clock to monitor the activity
  - Limit checker
  - Rise or fall time checker
  - Range checker
  - Frequency checker
  - Slew rate checker
- Analog Assertions

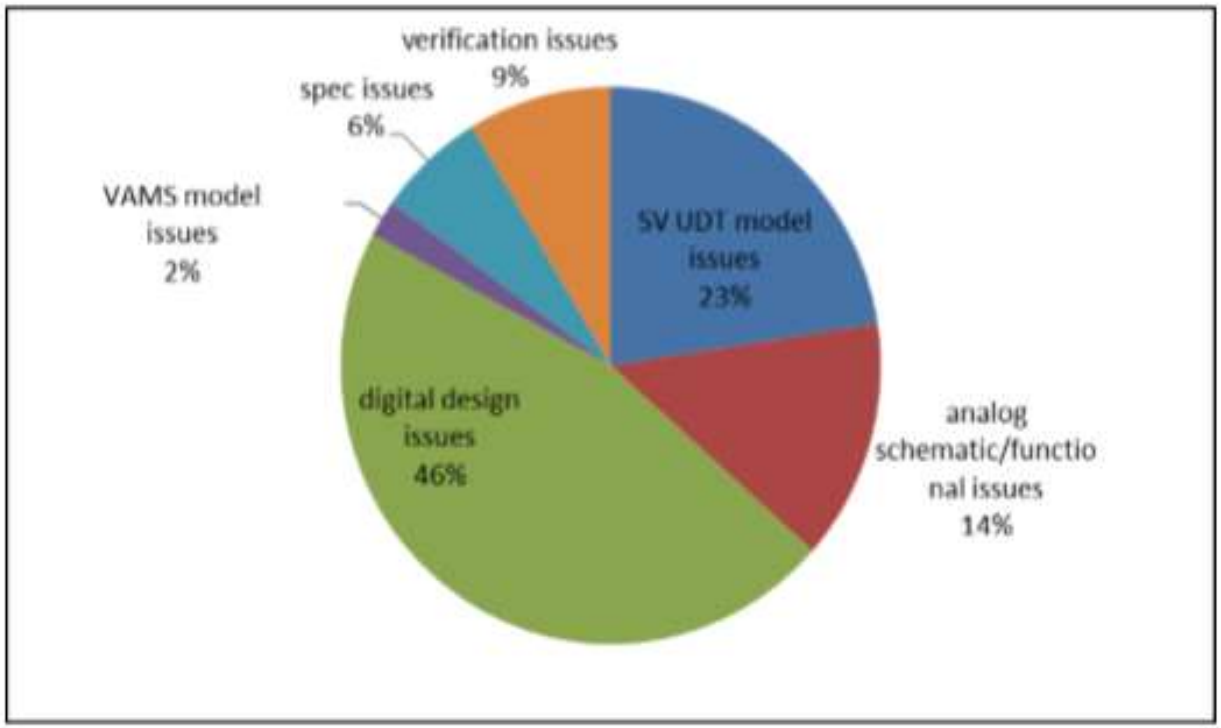
```
bit node_check_slew_rate;
always @(*) begin
 if($cds_analog_exists(signal_path) != 1)
  `uvm_fatal("AMS",sformatf("cds_analog_exists() failed for : %s",
signal_path))
 else begin
   vol_val = $cds_get_analog_value(signal_path);
   sr = 2 * 3.14 * freq *  vol_val;
   if(!(sr > sr_lo ) & (sr < sr_high)) begin
       node_check_slew_rate = 1;
      $assert("slew rate check failed …at %t",$time);
   end
 end
end
```

```
vnw_max_ANA_Core_1  assert dev="ANA_Core.*"  expr="V(n4)"
max=16.1 duration=1n message="[OV]VNW reverse biased"
level=warning anal_types=[dc tran]
vnw_min_ANA_Core_1  assert dev="ANA_Core.*"  expr="V(n4)"
min=-0.5 duration=1n message="[OV]VNW forward biased"
level=warning anal_types=[dc tran]
```

Regressions

- Nightly regressions for all tests with SV model
- AMS plan created targeting required scenarios
- Mixed VAMS and spice config selection as per DV plan
- VAMS regression was run on need basis
- Selected tests were run with full SPICE

Simulation Data



Conclusion

- Reusable scalable UVM env used for verification
- Realistic SV UDN model helped to run more system level scenarios
- Helped in quickly finding schematic issues
- More tests available to run on VAMS sims
- Mixed config speeds up sims but still uses SPICE for targeted module
- More time available for analog team to debug and run as tests resued
- Scoreboards, checkers reused for mixed signal sims