Challenges in Mixed Signal Verification

Amlan Chakrabarti

Sachin-Sudhakar Kulkarni

AMD

accellera SYSTEMS INITIATIVE™

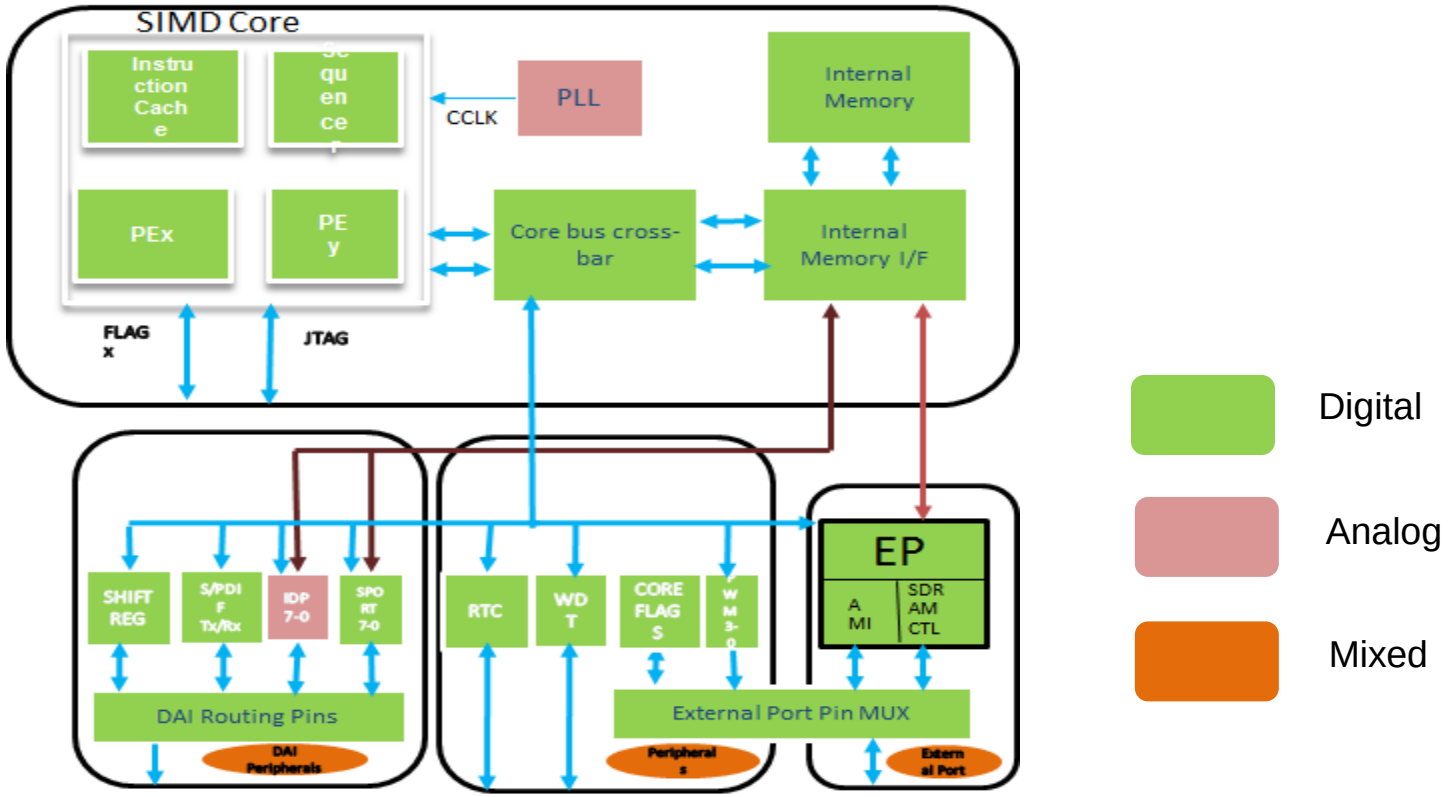
2015 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Agenda

- Introduction
- Challenges in mixed signal verification
- Prior work
- Proposed solution
- Results
- Conclusion
- Acknowledgements

Introduction

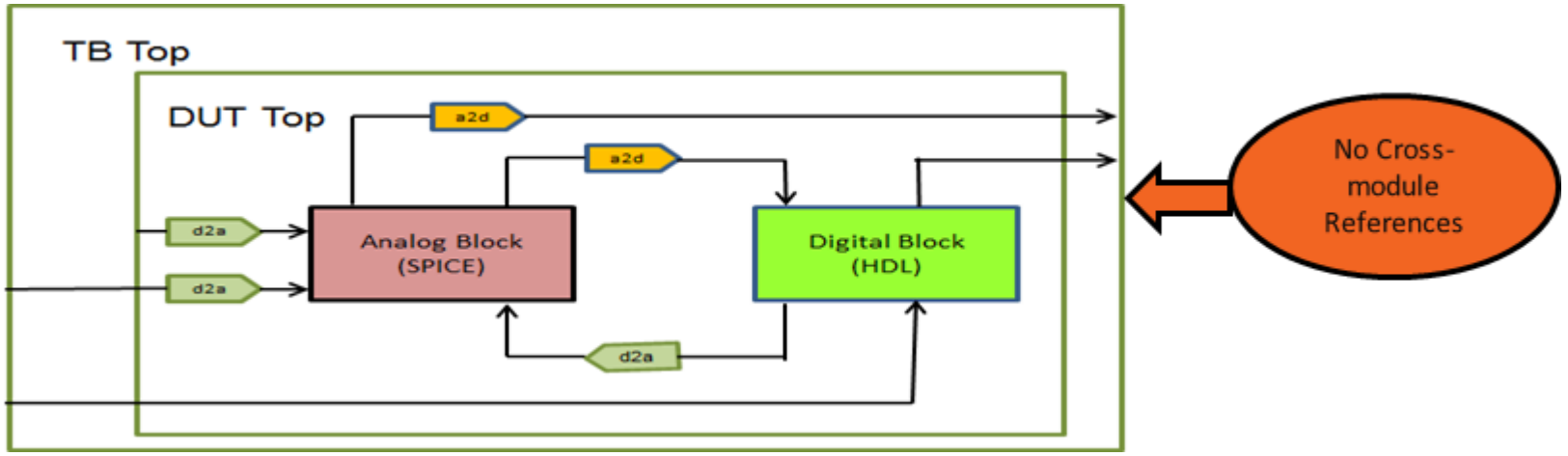
- Need to integrate more mixed signal components into SOCs.



Challenges in Mixed Signal Verification

- Challenges in mixed signal verification include –
- handling complex interactions between digital and analog circuits.
  - relationship between signal “value” and signal “strength” in both domains needs to be specified.
- long simulation time when simulating with a Spice netlist.
- creating system level failure scenarios at the IP level itself.
  - at the system level we need to know jitter margins of individual IPs.
- challenges in verifying PLLs.

Prior work in this field



- Prior work for handling interactions between digital and analog circuits
  - "Connect rules" developed to describe the signals which traverse the digital ->analog boundary and vice-versa.

Prior work (contd.)

- Default connect rules applied by the simulation tool
  - D2A rules

    (Digital) 0 → (Analog) 0V

    (Digital) 1 → (Analog) Local Supply voltage
  - A2D rules

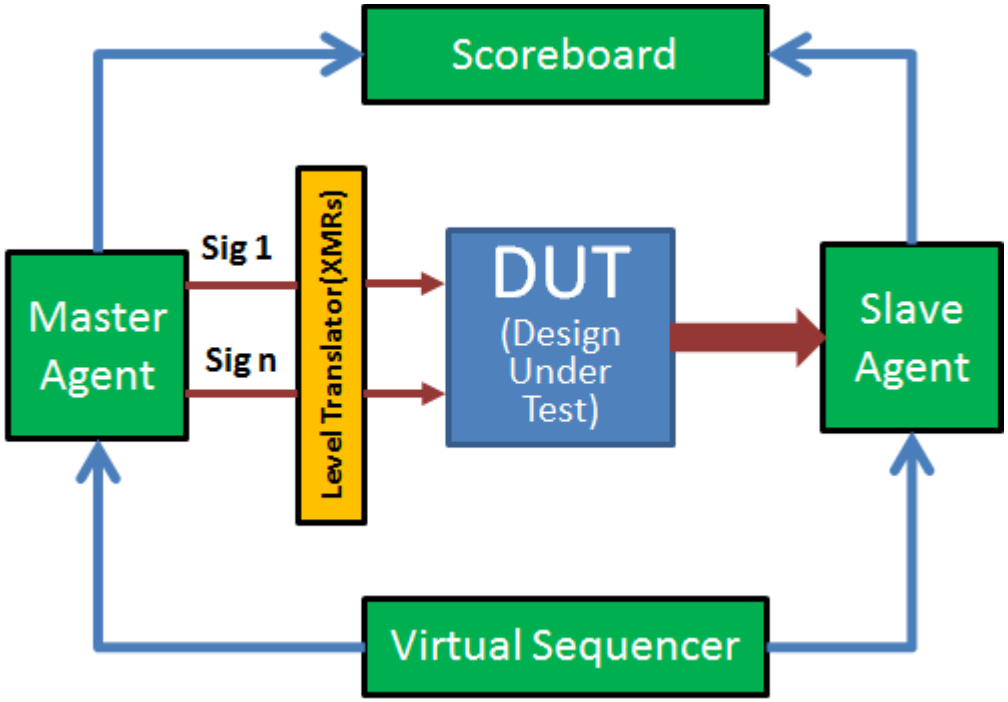
    (Analog) <=50% of Local supply voltage → (Digital) 0

    (Analog) >=50% of Local supply voltage → (Digital) 1
- Earlier mixed signal simulations run using a moderate/high accuracy for analog components.
- With many connect rules and a single accuracy setting for all analog components, simulations used to be slow.

Proposed Solution

- **Challenge of handling complex interactions between digital and analog circuits**
- Analog blocks with multiple-voltage domains, where supply voltage varies
  - D2a rule : d2a hiv=80% lov=20% node=<path to node> vdd=<supply_voltage_node>;
  - A2D rule : a2d loth=20% hith=80% node=<path to node> vdd=<supply_voltage_node>;
  - thresholds are a percentage of the supply voltage, not a fixed value
- Challenge in multi voltage analog blocks where the voltage level can vary dynamically during a simulation.
  - Supply voltage may remain constant.
- Developed a Level Translator block to handle this.

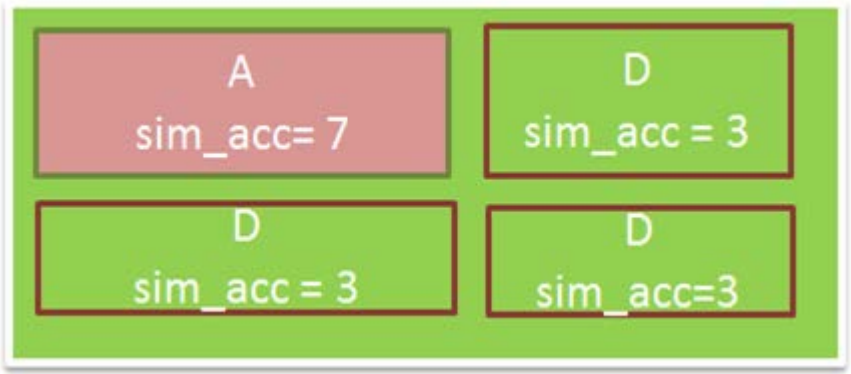
Level Translator



- Appropriate voltage is driven using Cross-Module References (XMRs) depending on the mode of operation.

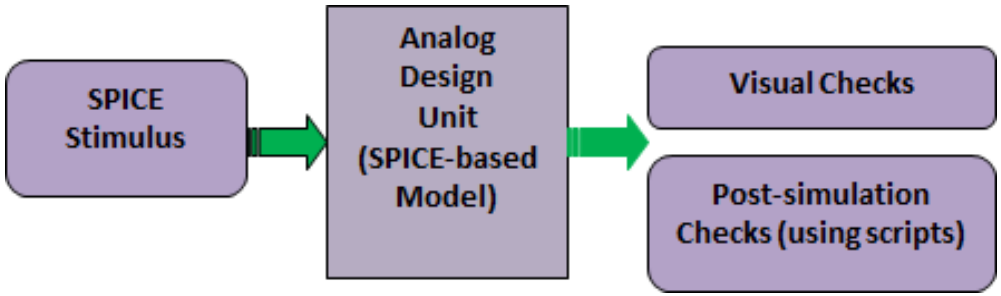
Long simulation time

- **Challenge of long simulation time when simulating with a Spice netlist**
- Use high accuracy for the analog units (slower simulation) and low/medium accuracy for the digital units(faster simulation)



Cross-Module References (XMRs)

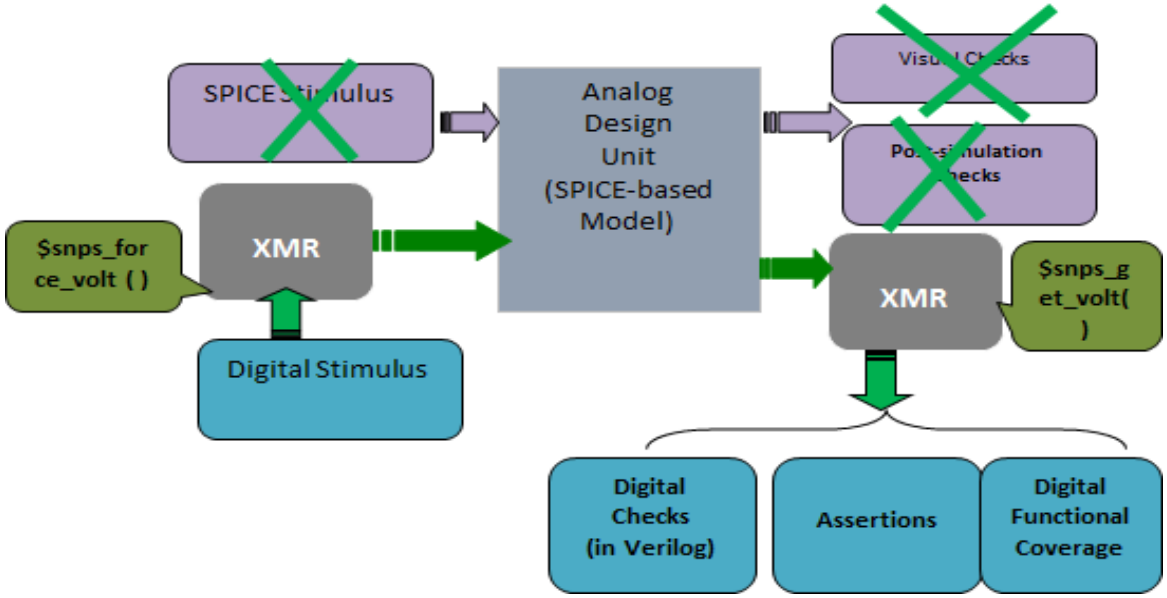
- Use Cross-Module References (XMRs) to reduce the usage of Connect Rules.
- Conventional verification flow for verifying an analog unit



- Stimulus applied to the analog unit from a Spice deck.
- Checking is through visual checking and scripting.

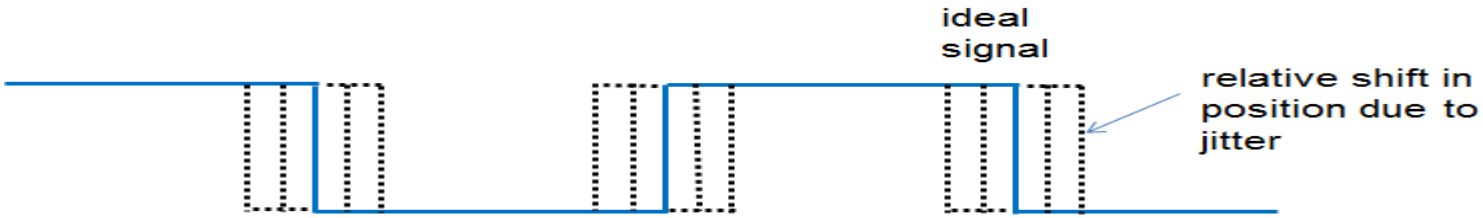
Cross-Module References (contd.)

- Through XMRs internal nodes in the analog view can be accessed from the digital view (testbench environment).
  - eliminate conversion between digital and analog domains



Creating system-level failure scenarios at the IP level

- **Challenge of creating system-level failure scenarios at the IP level**
- At the system level, it is important to know how much jitter can be tolerated by the individual IPs .
- If the jittered clock is generated from a PLL, it is difficult to control the amount of jitter generated.
- Can be achieved by replacing the PLL with a jitter model.
- ***What is jitter ?***



Jitter modeling

- In jitter model we modify the clock period or the width of a data bit driven from the clock.
- Types of jitter we have modelled –
  - **Deterministic (sinusoidal, triangular, etc.)**
  - sj_offset + (sj_ampl*$sin(2*3.1416*curr_sj_freq*$realtime));
    where
    sj_offset = an initial offset of the jitter sinusoid
    sj_ampl = amplitude of the jitter sinusoid
    curr_sj_freq = frequency of the jitter sinusoid
  - **Random (Gaussian)**
  - rj_offset + $dist_normal(rj_seed, 0, rj_stdev)*10/1000;
    where
    rj_offset = an initial offset for the random jitter
    rj_seed = initial seed for the normal distribution
    rj_stdev = standard deviation for the density function
  - **Combination of deterministic and random**
- Gives an idea of the data eye.

Challenges in PLL verification

- **Challenges in verifying PLLs**
- If the PLL has an LC oscillator, in simulations the LC oscillator won't toggle after power supplies come up.
- Solved by adding a kick to the LC tank by forcing some voltage on the LC tank nodes at the start of the simulation
  - ***$snps_force_volt (testbench.pll_inst.lcpll.tank_bias, 1.2);***
- Long ramp-up times for Bandgap and regulator
  - solved by using a separate simulation for Bandgap/Regulator ramp-up.
  - for all other operational modes, force the bandgap reference voltage and regulator supply voltages.
  - ***$snps_force_volt(<node_name>) -voltage <in volts> –time <time units>***

Challenge in VCO modeling

- **Challenge in developing an accurate model of the VCO**
- In SystemVerilog, VCO can be modeled using real numbers.
  - avoids the need to use Verilog-A or Verilog-AMS models.
- Code snippet

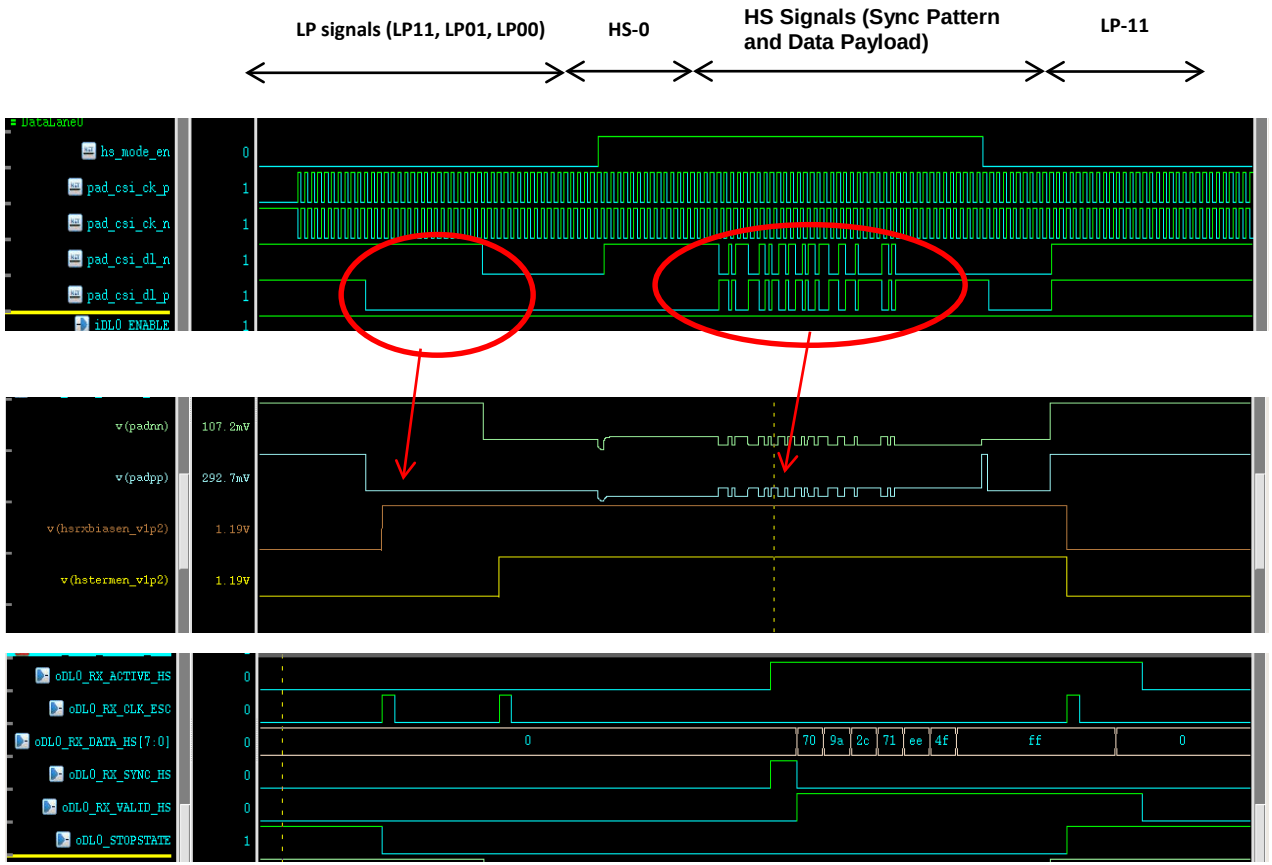
```
output wire vco;
real  minfreq,maxfreq,currentfreq;
real minvolt = 0.3;real maxvolt =0.9;
parameter ghz = 1e+09;
always@*
  begin minfreq =  0.9504 * ghz ; maxfreq =  8.084 * ghz ;
  end
always @*
    gain = (maxfreq – minfreq)/ (maxvolt –minvolt);
real fctl,halfperiod;
```

VCO model using SystemVerilog

```
always @*
    if (vctl < maxvolt && vctl > minvolt)
        fctl = (vctl – minvolt)*0.25*gain;
    else if(vctl >= maxvolt)
         fctl = (maxvolt-minvolt)*0.25*gain;
    else if(vctl <= minvolt)
         fctl = 0.25*gain;
 always@*
    currentfreq = fctl +minfreq;
always @*
   halfperiod = (1000*ghz)/(2*currentfreq);
initial vcoPre = 0;
//Generate clock
always
    #halfperiod vcoPre = ~vcoPre;
assign vco =vcoPre;
```

Results

- Waveforms with the Level Translator block



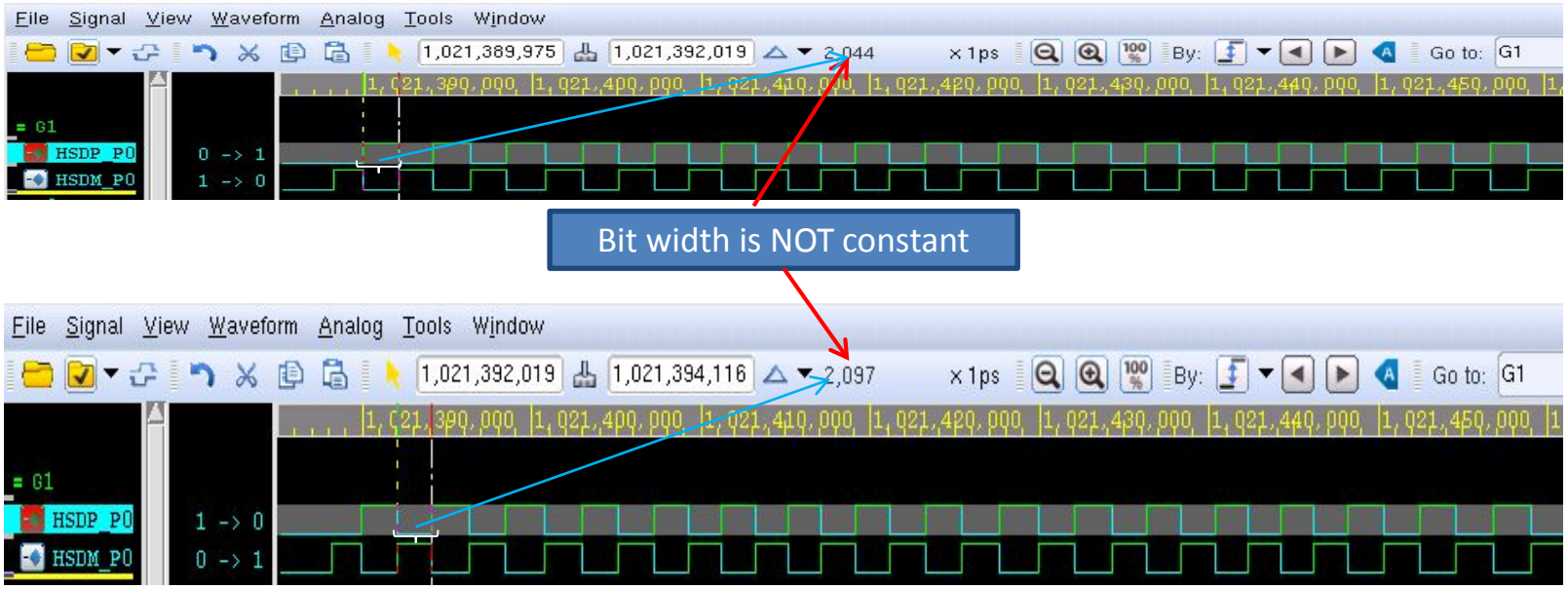
Stimulus from the Digital UVM testbench

Actual LP and HS voltage levels at the PADs **[“Converter”]**

De-serialized data out of the CSI2 Rx-PHY

Results (contd.)

- Results for PLL Spice simulations using targeted accuracy settings and Cross-Module References
  - closed loop simulations complete in 4-5 hours
  - earlier simulations used to run for 16-18 hours
- Introduction of jitter on data on a serial interface



Conclusion

- Mixed signal implies blocks which have both analog and digital content.
- Significant challenges need to be solved to reap maximum benefits from mixed signal verification.
- To handle complex interactions between digital and analog circuits, a level translator block may be needed.
- Long simulation time when simulating with a Spice netlist can be solved using targeted simulation accuracy settings and cross-module references (XMRs).
- Replacing the PLL with a jitter model enables to control the amount of jitter generated.

Acknowledgements

- We would like to acknowledge our colleagues for their valuable suggestions during this effort
  - Ratheesh Mekkadan
  - Deepa Ananthanarayanan

Thank You !

Questions ?