2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Practical Considerations for Real Valued Modeling of High Performance Analog Systems

Dushyant Juneja, Analog Devices, Inc.

Siddharth Prabhu, Analog Devices, Inc.

Syam Veluri, Analog Devices, Inc.

ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Agenda

- Introduction
  - Behavioral Modeling
  - RVMs – Introduction and flavor comparisons
- Modeling aspects in dealing with RVMs
- Integration aspects
  - Netlisting & Validation
- What's missing, exactly?
  - With the standard
  - With the tools
- Conclusion

Real Valued Modeling

- *Behavioral modeling technique* to simulate *analog functionality* within a *digital simulation*
  - Required for robust Universal Verification Methodology (UVM), Metric Driven Verification (MDV)
  - Event-driven evaluation → faster top-level AMS functional verification environment
- SystemVerilog offers new constructs - user-defined nettypes, (UDTs) & user-defined resolution functions (UDRs).
- RVM augments other detailed models, may not completely replace them!

Real Valued Modeling

- Speed over accuracy and modeling details captured
- Abstraction level is much higher closer to sub-system level blocks

```
// Amplifier gain
real gain = 1000 ;

// Output computation
always @( inp, inm, vdd, vss ) begin
  tmp_in = inp - inm ;
  tmp_out = gain * tmp_in ;

  // Hard limiting the output
  if (tmp_out > vdd) : tmp_out = vdd ;
  else if (tmp_out < vss) : tmp_out = vss ;

  #1 out = tmp_out ;
end
```

Comparison with Behavioral Modeling Standards

Approach	Verilog-A	Verilog-AMS	VAMS wreal	SystemVerilog (2012)
Description	Verilog-lookalike for SPICE	Superset of Verilog-A and Verilog-D	Verilog-AMS RVM subset	Most recent approach for RVM
Evaluation	Continuous	Flexible	Discrete time	Discrete time
Digital Integration	Via Cosims	True AMS, limited UVM/SV	True AMS, limited UVM/SV	Excellent integration
Modeling Features	True analog modeling	True analog + mixed signal interaction	Abstract signal chain only	Abstract signal chain, but more permissive
Speed*	Slow	Better	Close to digital	Fast, close to digital

*Speed is a subjective matter –
the comparison shown is approximate and highly dependent on level of abstraction.

RVM Feature Comparison

Feature	SV 2009 Real	Verilog-AMS wreal	SV 2012 Real
Real values	Supported	Supported	Supported
Real Nets, Unidirectional ports	Supported	Supported	Supported
Bidirectional real ports	Not supported	Supported	Supported
User defined nettypes and Resolution	Not supported	Restricted	Supported
Real Randomization & Constraints	Not supported	Not supported	Limited support
Coercion	Not supported	Supported, but patchy	Using generic ‘interconnect’ type

SV 2012 RVM – UDTs/UDRs

- RVMs – promising approach for mixed signal verification
- SV 2012 the 'latest & greatest' of the suite
  - Extends RVM methods via 'nettypes' (my_type in code)
  - Custom resolution
  - Single kernel simulation
- Several new concerns!!!

```
// user-defined type my_type
typedef struct {
            real val;
            bit enable;
} my_type;

function my_type mtsum(input my_type
driver[]);
 mtsum.field1 = 0.0;
 foreach (driver[i])
  mtsum.val +=driver[i].val;
endfunction
```

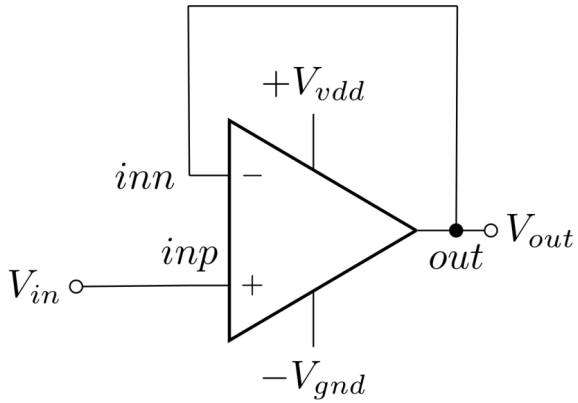
```
module my_mod (...);
output OUT;
input IN;
nettype my_type IN with mtsum;
...
```

Analog Event Explosions

Simple non-inverting amplifier configuration.

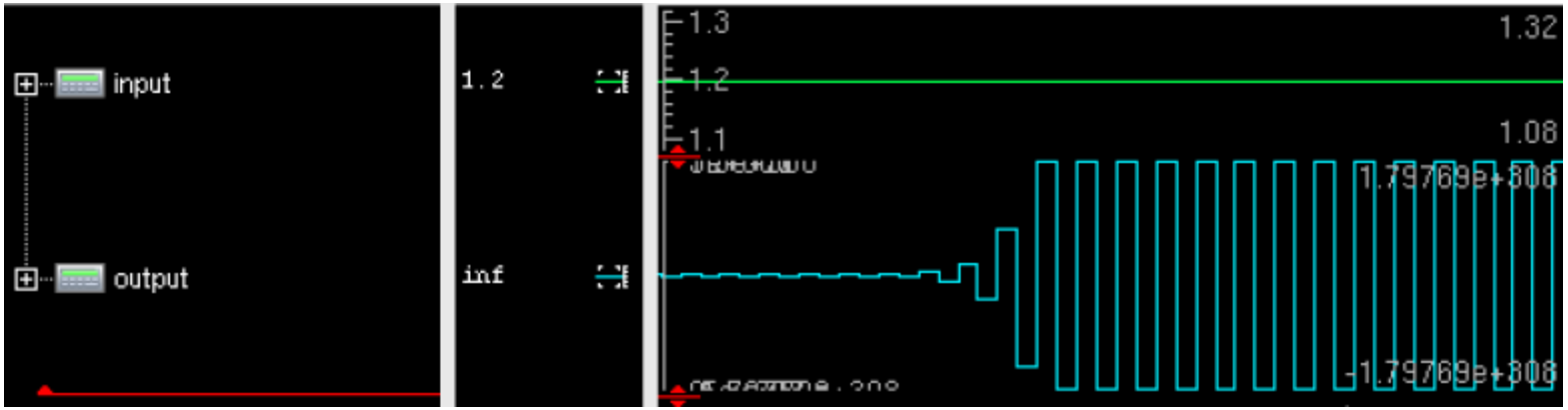
Expected:

$$V_{out} = V_{in}$$



Actual:

Unstable oscillations!!!



Analog Event Explosions

Possible Issues:

- **Zero Delay Feedback** on finite delay amp → corrupt output!

- Unconditionally unstable configuration

$$H_{expected\ (opamp)} = G, G \gg 1$$

$$H_{actual\ (opamp)} = \frac{G}{1 + G \cdot z^{-1}}$$

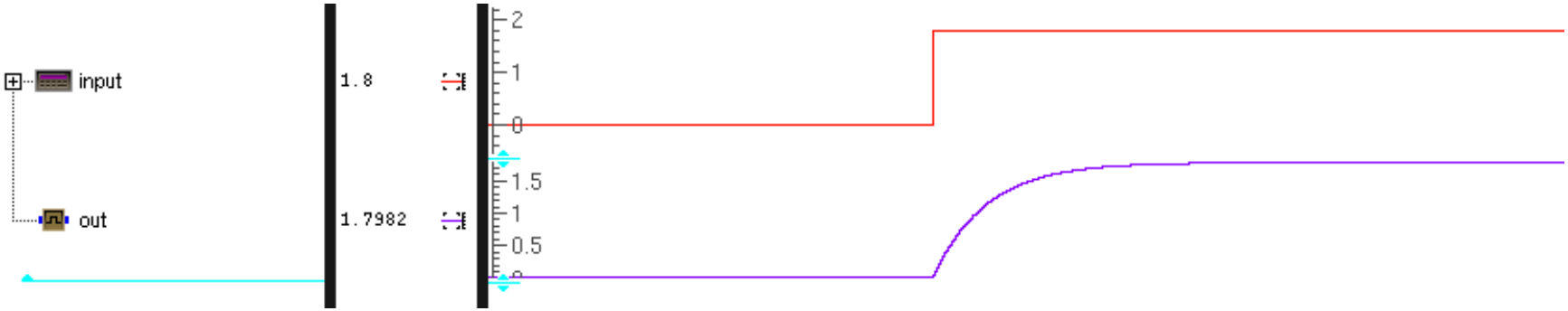
Analog Event Explosions

Solution:

Can be compensated by Amplifier pole (band limited amplifier), if:

$$\delta < \tau/G$$

δ = simulator time step, τ = Amplifier time constant



Application – Simulation timestep needs to be carefully chosen!

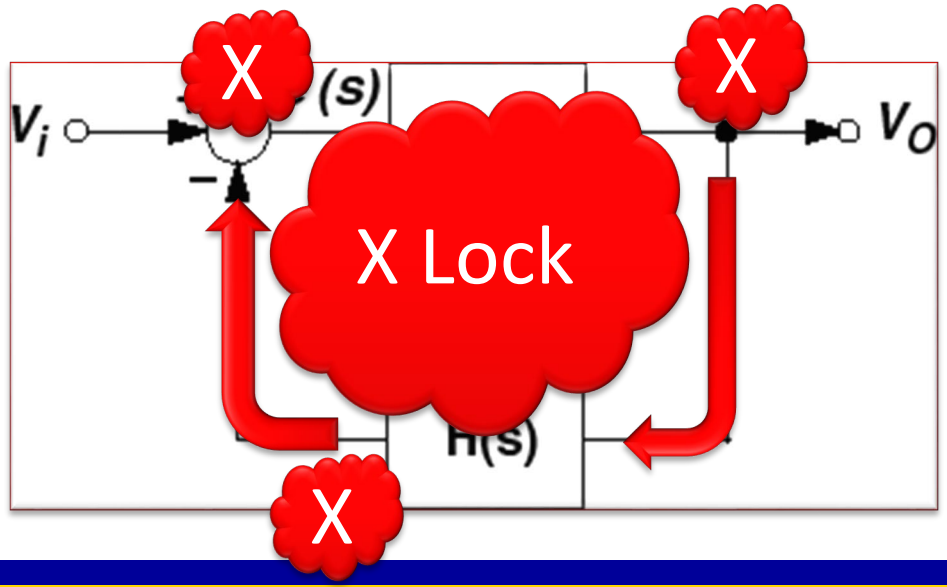
Analog Event Explosions

Other Concerns:

1. Complex Configurations
    1. Multiple drivers on node such as ‘inn’ → Sorted out by custom resolution functions
    2. Impedance based feedbacks → Need for custom nettypes
2. Artificial error → Needs to be tailored using timestep control
3. Too many features, tight timestep → Slow simulations!

Analog Contention & Unknown States

- Multiple driver resolution and Unknown states – vitally important for AMS modeling
  - Improperly handled digital unknown states can shadow real bugs!
  - Mixed signal contention on muxed IOs
  - Contention due to testbench drivers
- Example – Feedback loop
  - X-Lock corrupts the loop!



Analog Contention & Unknown States

- Sorted out via nettypes:

```
 1 typedef struct{
 2   real V;    // Voltage value
 3   real I;    // Current flow value
 4   logic isX; // flag for detecting unknown voltages
 5 } my_net ;
 6
 7 // Resolution function to highlight unknowns
 8 function my_net my_res_fnc(input VI_with_X node[]);
 9   logic result_is_X ;
10   int i ;
11   ...
12   for ( i=0 ; i < node.size() ; i++ ) begin
13     if ( node[i].isX === 1'b1 )
14       result_is_X = 1'b1 ;
15     ... // Other stuff for dealing with V and I
```

High Impedance and Ground References

- Extremely common in AMS IPs.
- Need a way to determine ‘strength’ of a ‘node’/driver, and resolve accordingly
  - Better still, resolve based on ‘drive impedance’
- Nettypes come to the rescue!
  - Can be extended to handle impedances
  - Scalable approach

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

High Impedance and Ground References

```
 1 // New user defined type
 2 typedef struct {
 3   real voltage; // The actual signal
 4   real Zr; // Real component of impedance/strength
 5 } my_udt ;
 6
 7 function my_udt my_res ( input drivers [] ) ;
 8   ...
 9   for ( i=0 ; i < drivers.size() ; i ++ )
10     if ( Zr = 0 ) // Ground/Strong Reference case
11       final_v = drivers[i].voltage;
12     else if ( Zr > Zr_max ) : next ; // Driver can be ignored!
18     ...
```

Integration Concerns

- Netlisting - Stitching/Connecting RVMs as per schematics (Structural concern)
  - Deriving top level SV netlist from analog repo
  - Mixed signal buses
  - Inherited connections
- Integration with top level digital sims
  - Coercion
  - Nettype compatibility
- Analog stimuli generation

RVM Netlisting – Mixed Signal Buses

Mixed Signal buses – various concerns

- No SPICE-like bitwise connections
  - Difficult to figure out connectivity at a glance
- AMS Buses typically have unpacked assignments
  - Disturbs majority SV netlisting streaming tools (output packed arrays by default)
  - Kills scope for bus concatenation
  - Bus reversal, etc. not possible either
- Need to be worked around by flattening and removing bus references

RVM Netlisting – Mixed Signal Buses

```
module mod_name(
            inout my_net ibus [3:0] );
...
endmdoule
```

```
mod_name inst_name ( .ibus(obus) ) ;
```

```
mod_name inst_name (
.ibus(`{p1,p2,p3,p4}) ) ;
```

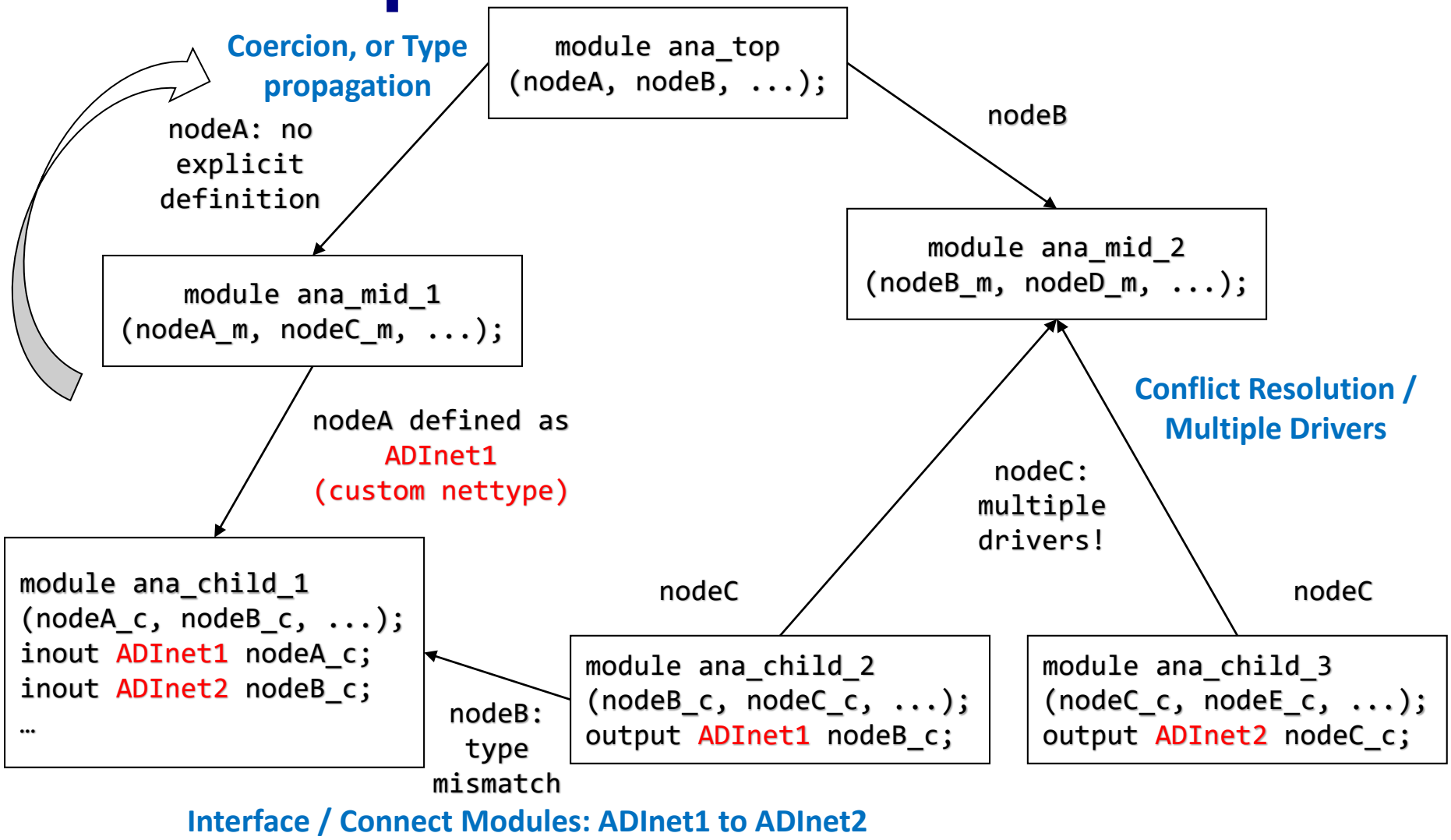
```
mod_name inst_name ( .ibus[0](p1),
ibus[1](p2), . . . ) ;
```

RVM Netlisting – Inherited connections

Inherited connections – kind of global pins, inherited from higher levels in schematics

- Avoids cluttering the design symbols with repeated pins
  - Analog designers use it to make pretty schematics
- Actually just another invisible connection
- Majority of vendor tools fail to netlist this properly, need to be worked around with custom scripts

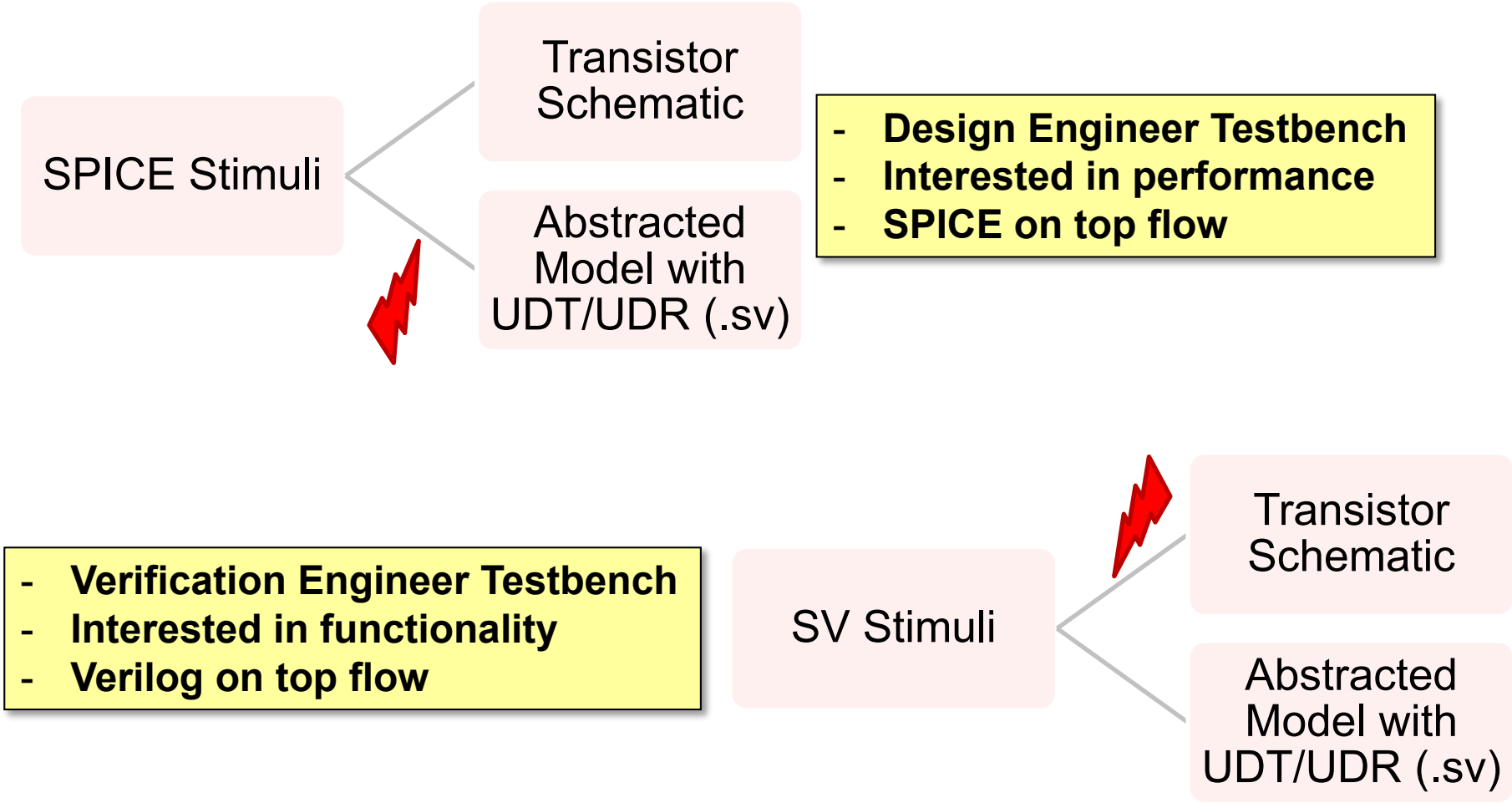
RVM Netlisting – Key requirement?



RVM Netlist - Coercion and Conflict resolution

- *NOT* an issue for just digital simulations or single nettype simulations
  - `` `default_nettype `` fails to handle multiple nettype scenario
- Need a forced resolution on nets of different discipline
  - Multi-driver, mixed discipline nettypes
  - Real – logic and arrayed ports
  - doing conflict resolution too
- Generic '`interconnect`' to the rescue
  - Manual edits may be needed for tool generated netlists
  - What coerced to what? Still a mystery… !

RVM Simulation and Model Validation – connect rules



RVM Integration – Other Concerns

- Not friendly with Low Power Intents like CPF/UPF…
  - Coercion breaks!
  - Power aware connect modules?
- SV Interfaces not too friendly with RVMs/nettypes
- Schematic capture tools – new checks needed (like SV keyword checks)
- Real randomized/Constraint Randomized stimulti!

…

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Next Steps & Conclusion

- Discussed practical issues and work-around in Real Valued Modeling of High Performance Analog
- Limitations faced from IEEE 1800-2012 (SV-LRM)
  - Critical improvements required
- In spite of all these issues, we still believe SystemVerilog 2012 along with MDV/UVM flow will greatly enhance productivity.

- Will standards committee and EDA tools step up for resolving the challenges?!

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Next Steps & Conclusion

- Develop a parameterized library of Real Valued Models
- Automated Insertion of Connect Modules
  - Looking forward to SV-AMS
  - Validation flows
- Top-Level SystemVerilog digital netlisting
- EDA tools, Standards committees and Verification teams have scope for mutual collaboration

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Thank You