# Practical Considerations for Real Valued Modeling of High Performance Analog Systems

Dushyant Juneja, Siddharth Prabhu, Syam Veluri
Analog Devices, Inc.

## Real Valued Models (RVMs)

Digital Friendly Behavioral Modeling Methodology, intended for Mixed Signal Functional Verification

Example Amplifier Model:

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

// Output computation
always @( inp, inm, vdd, gnd ) 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
```

Pros:

- ✓ Tight Integration with Digital/SystemVerilog
  - → UVM, MDV work out of the box!
- ✓ Improved feature set with IEEE 1800:2012 LRM
  - → Helps move beyond signal chain abstraction
  - → Generic interconnect gives improved coercion
- ✓ Inherently Event Driven
  - → Excellent Simulation Speed!

Cons:

- × Event-loop explosion
- × Integration Concerns: Netlisting, conflict resolution, coercion
  - → Increased manual effort
  - → Mixed Signal Buses
- × Analog Stimuli concerns
  - → Randomization, etc. do not work out-of-the-box

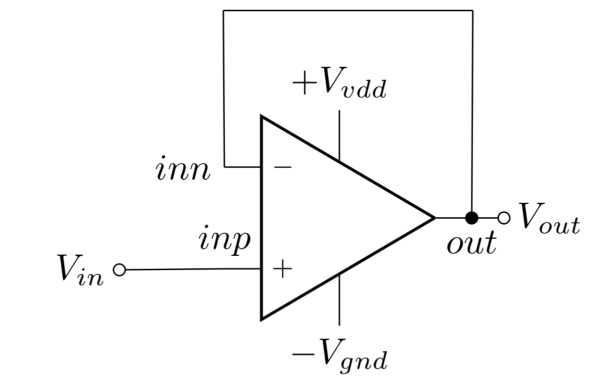
Comparison with Other Methods:

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

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

## Analog Modeling – Zero Delay Feedbacks

Figure 1




Simple non-inverting amplifier configuration. Expected output voltage/transfer function:

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

However, this gets difficult to model:

1. Node 'inn' could have multiple drivers for complex configurations → Sorted out by custom nettypes (SV 2012)
2. Zero Delay Feedback → Simulation gets hung
3. Small, Artificial Delay → Amplifier gets unstable (Figure 2)

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

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

1. Time step sensitive pole.
2. Unconditionally unstable for all practical opamp gains

**Solution:**
Can be compensated by Amplifier pole (bandlimited amplifier), if:

$$\delta < {}^{\tau}/_{G}$$

$\delta$ = simulator time step, $\tau$ = Amplifier time constant

**Application** – simulation time step needs to be carefully chosen!

Figure 2: Artificial instability in outputs

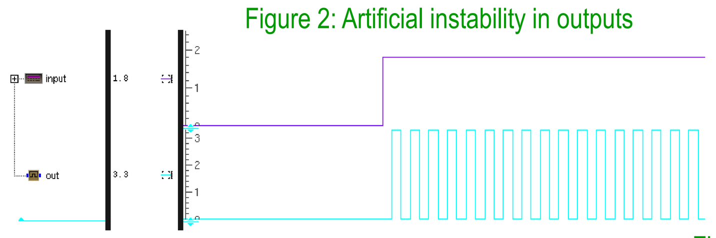
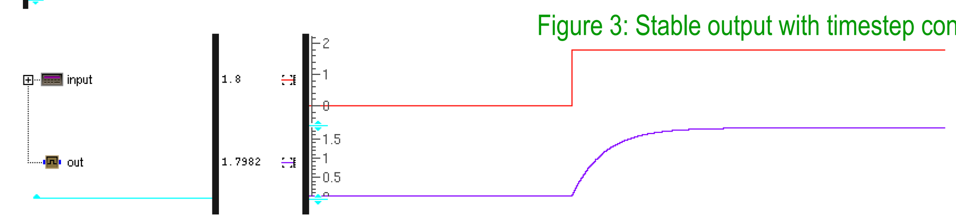


Figure 3: Stable output with timestep control



## Analog Stimulus Development

Analog Stimuli are quintessential for mixed signal testbenches. The stimuli need to be able tunable to assist custom protocols, and have the ability to drive mixed signal randomized vectors. However, there is limited support for 'real' randomization amidst vendor tools, and sometimes even requires special licenses. This was worked around by suitably constrained random integer division:

```
class xtndA;
  rand int a; // Universally supported construct
  constraint cst1 { a < 1230; }
  constraint cst2 { a > 0; }
endclass
 ...
if(ca.randomize() == 1) begin
  x = ca.a/123.0;
  $display("REAL Value = %g \n", x);
end
```

## Analog Modeling – Contention & Unknown states

High Impedance and Unknown states – vitally important for AMS modeling. Example scenarios:

1. Mixed signal feedback loops. Improperly handled digital unknown states can shadow real bugs! (Figure 4).
2. Mixed signal contention on chip pads/block pins

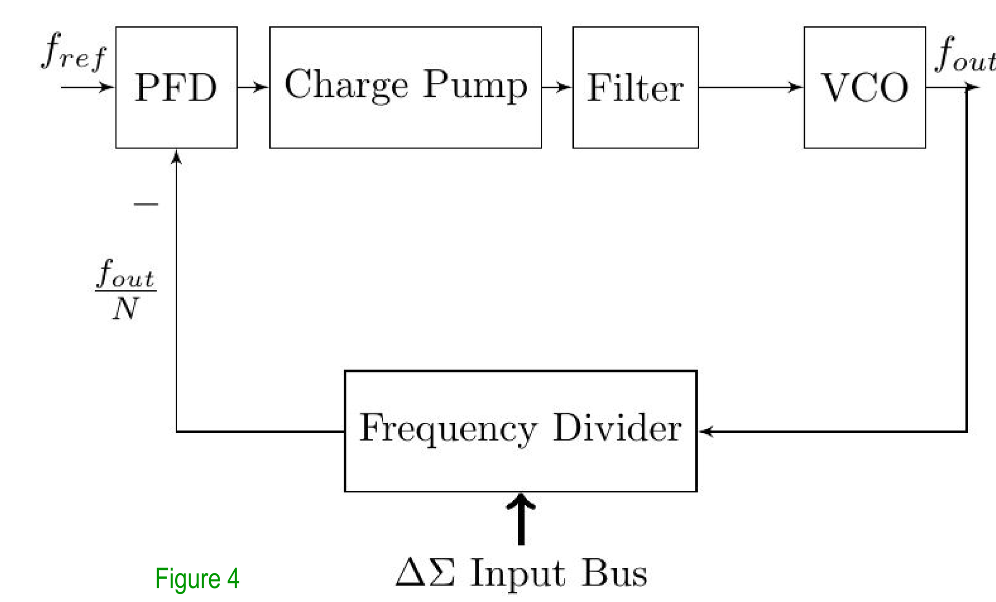



Figure 4

Achieved via nettypes:

```
typedef struct{
  real V;    // Voltage value
  real I;    // Current flow value
  logic isX; // flag for detecting unknown voltages
} my_net ;

// Resolution function to highlight unknowns
function my_net my_res_fnc(input VI_with_X node[]);
  logic result_is_X ;
  int i ;
  ...
  for ( i=0 ; i < node.size() ; i++ ) begin
   if ( node[i].isX === 1'b1 )
     result_is_X = 1'b1 ;
   ... // Other stuff for dealing with V and I
endfunction
```

## Analog Modeling – High Impedance and Ground Refs

High impedance and Ground references – Extremely common in AMS IPs.
Traditional methods do not allow 'strong' or 'weak' real nets. Even if this was allowed, the limited array of options may not be sufficient. Nettypes help sort out this situation by allowing "drive impedance" on nodes. Appropriate updates need to be done in resolution functions as well to mimic this function as required.

```
// New user defined type
typedef struct {
  real voltage; // The actual signal
  real Zr; // Real component of impedance/strength
} my_udt ;

function my_udt my_res ( input drivers [] ) ;
  ...
  for ( i=0 ; i < drivers.size() ; i ++ )
    if ( Zr = 0 ) // Ground/Strong Reference case
      final_v = drivers[i].voltage;
    else if ( Zr > Zr_max ) : next ; // The driver
can be effectively ignored!
    ...
```

## Structural Concerns

**Netlisting** and **Integration** are two major concerns here. The following challenges were faced in this course:

1. Need for SV 2012 netlister.
2. Type coercion.
3. Mixed Signal Buses.

The netlister was worked around using a custom script.

**Type Coercion** refers to type-propagation for leaf level nets all the way to the top level. This is necessary because the top level connections remain essentially 'typeless' otherwise, causing compile issues. A corollary is **conflict resolution**, where a conflict between multiple nettypes needs to be sorted out. A pictorial representation is as in Figure 5. The coercion was carried out using SV 2012's generic `'interconnect'`, while conflict resolution was sorted out manually for the current flow.

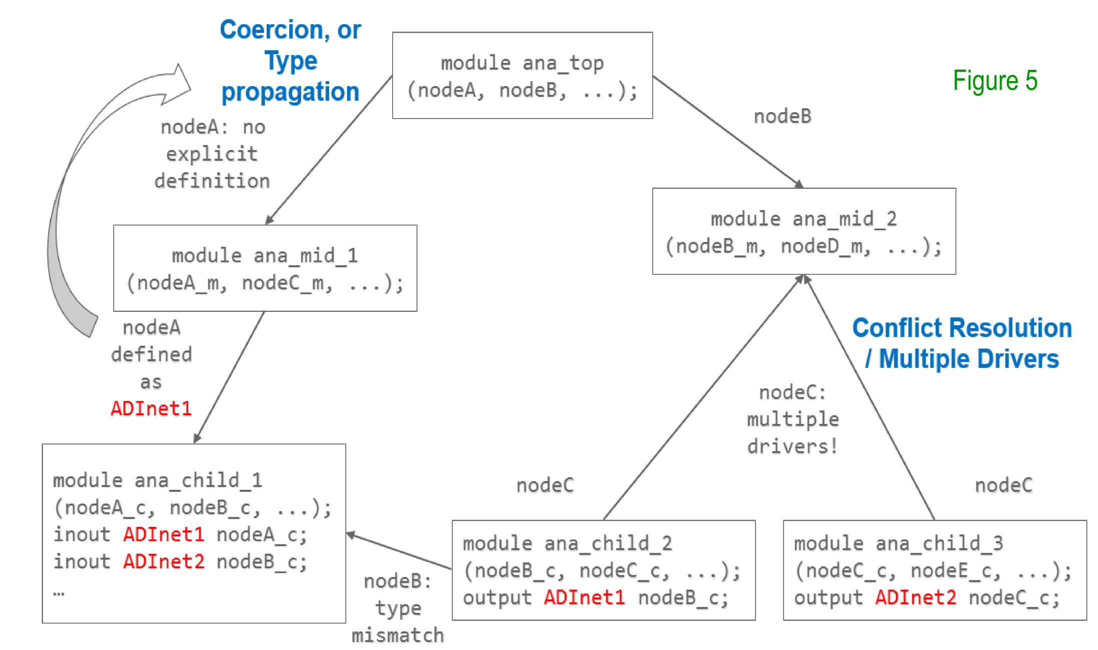



Figure 5

**Mixed Signal Buses** pose another major problem for the integration. While structurally supported, buses cannot be connected bitwise or even part-wise. Further, unpacked buses created issues when present at non-leaf level hierarchies (such as top level or block level), and were a pain to deal with. For the current flow, these buses were exploded to individual ports.

## Results

The zero delay feedback usage workarounds suggested were benchmarked for simulation accuracy and speed for the case of non-inverting amplifier circuit. As observed, the ideal amplifier gotcha actually slowed down the simulation, even for the case where it did come out of delta cycles (that is, where explicit delay is provided in the model). On the other hand, after modifications as discussed, the simulations were ~100x faster!

| Modeling | Speed | Accuracy |
|---|---|---|
| VerilogA/SPICE | 1x (Reference) | Reference |
| RVM with ideal amplifier | 0.02x | Not converged |
| Bandlimited RVM | 100x | 1% error |

## Acknowledgements

1. Lakshmi Narayanan, Dave Smart – Analog Devices
2. Cadence Design Systems