DVS Interface Element—A Novel Approach in Multi-Power Domain, Mixed-Signal Design Verification

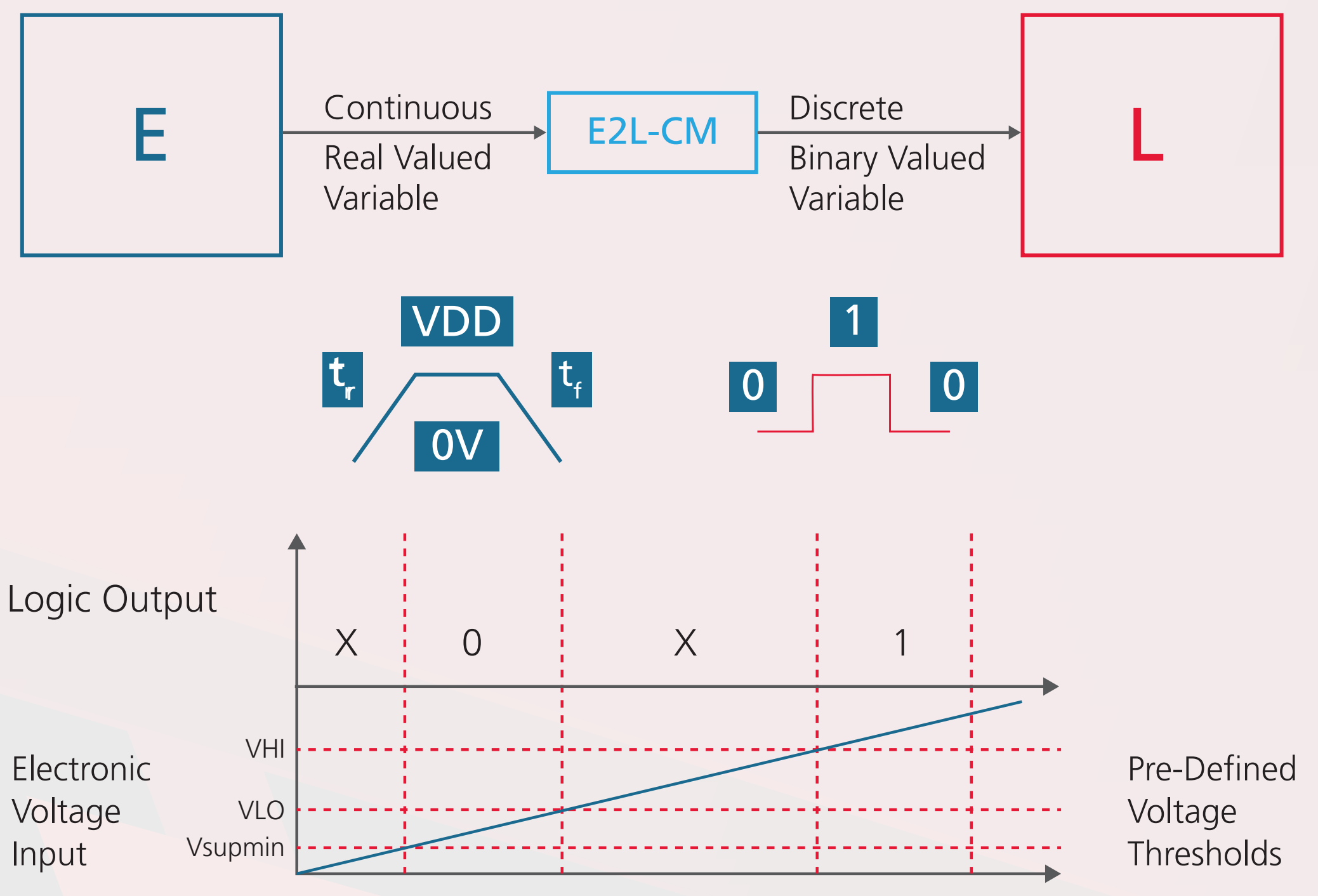
With a closer look at its emergence and holistic features

Shekar Chetput, Software Engineering Director, Cadence

Challenging aspects in AMS verification

- Accurate verification methodology - Takes care of:
  - Domain crossings in a **power-managed design**
  - Inter-discipline signal traversal
  - Circuit conditions or states (on/off), etc.
- Has dependency on the IP owner
  - Verification engineer's lack of power domain knowledge at subsystem level
- Power domain specification
  - Not a trivial task
  - Inaccuracy leads to **functional failure**

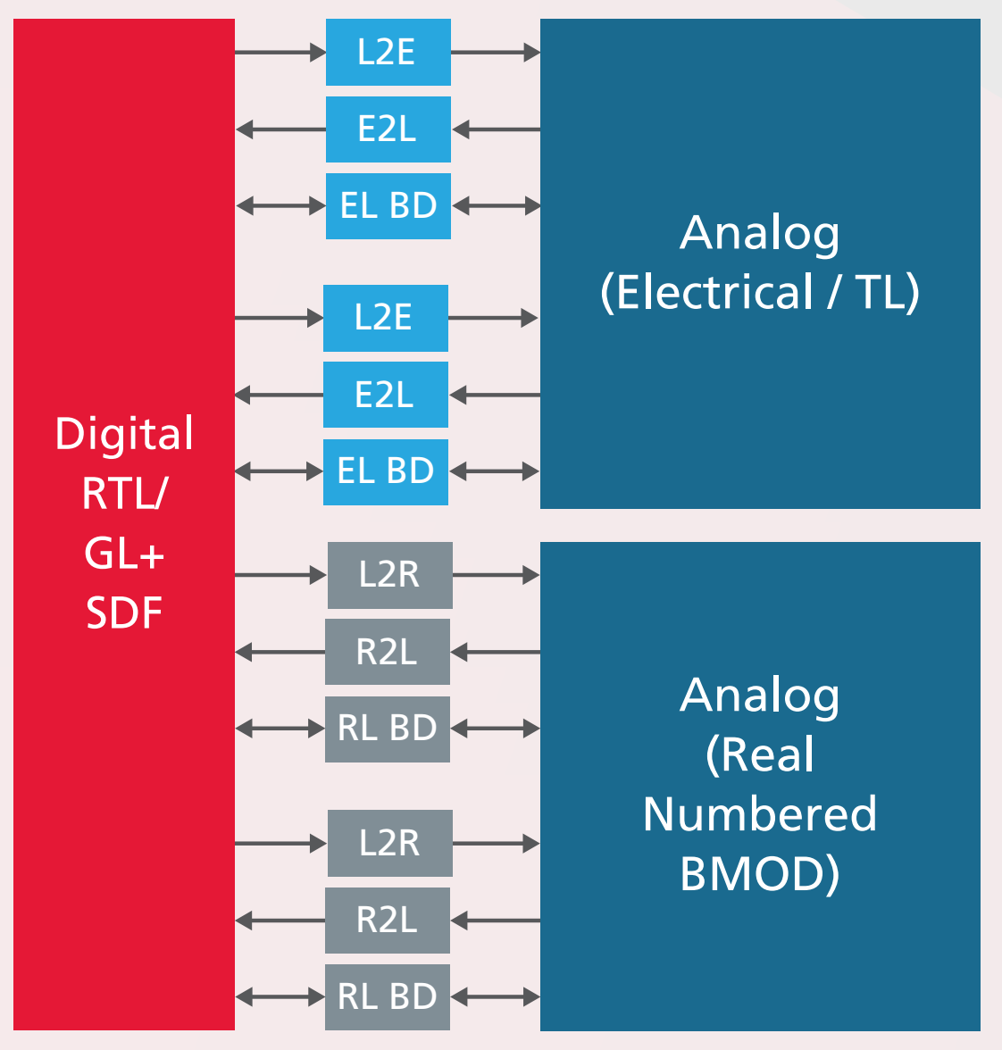
Connect modules for AMS and DMS



Static connect module

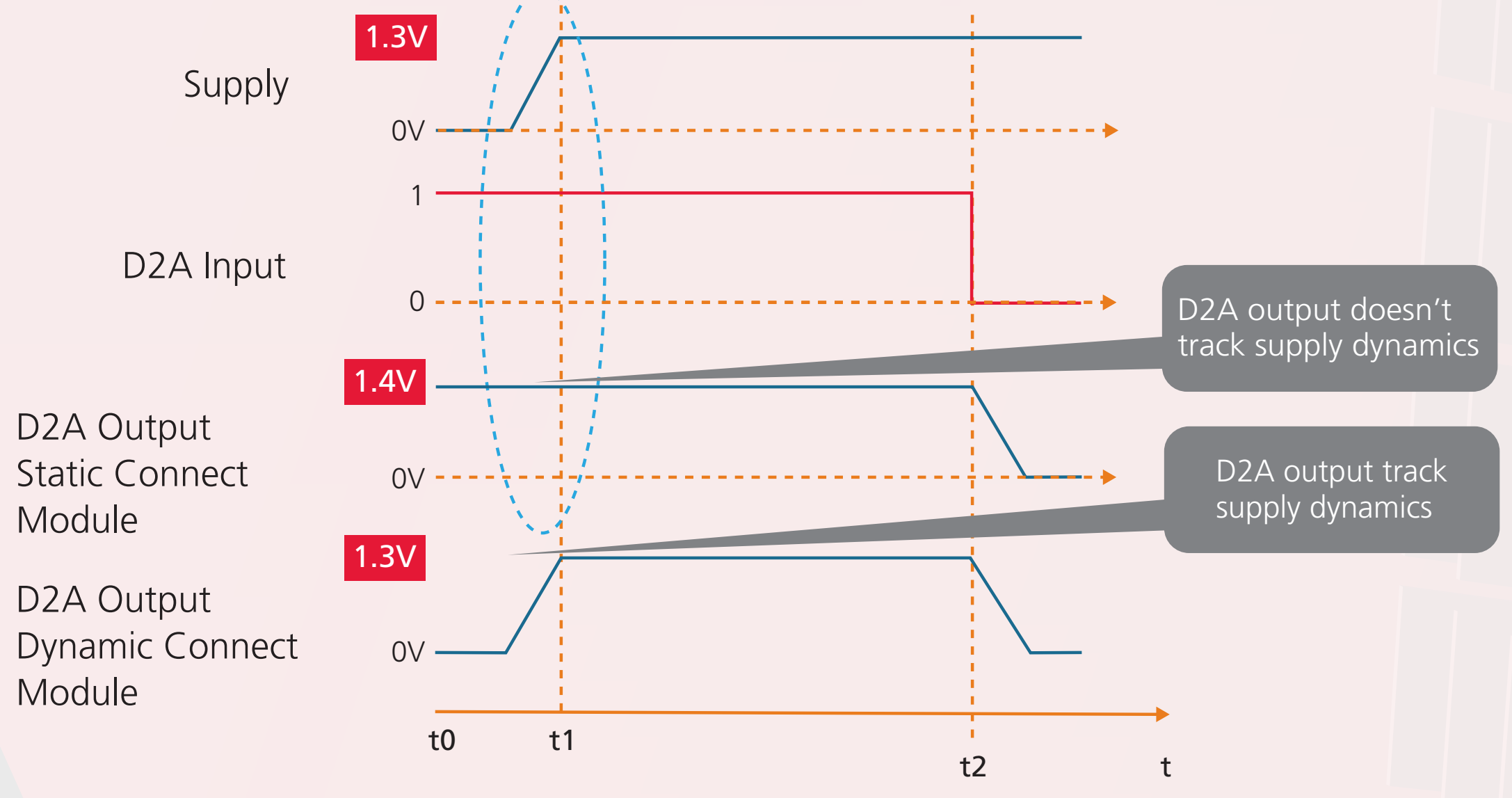
```
amsd {
ie vsup=1.2
cell = “digtop”
inst = “top.ana.buff0”
}
```

- Manual voltage domain handling
  - **No dynamism**
  - **Fixed supply voltage levels per simulation run**



Dynamic connect module

- Very critical with multiple voltage levels and supplies

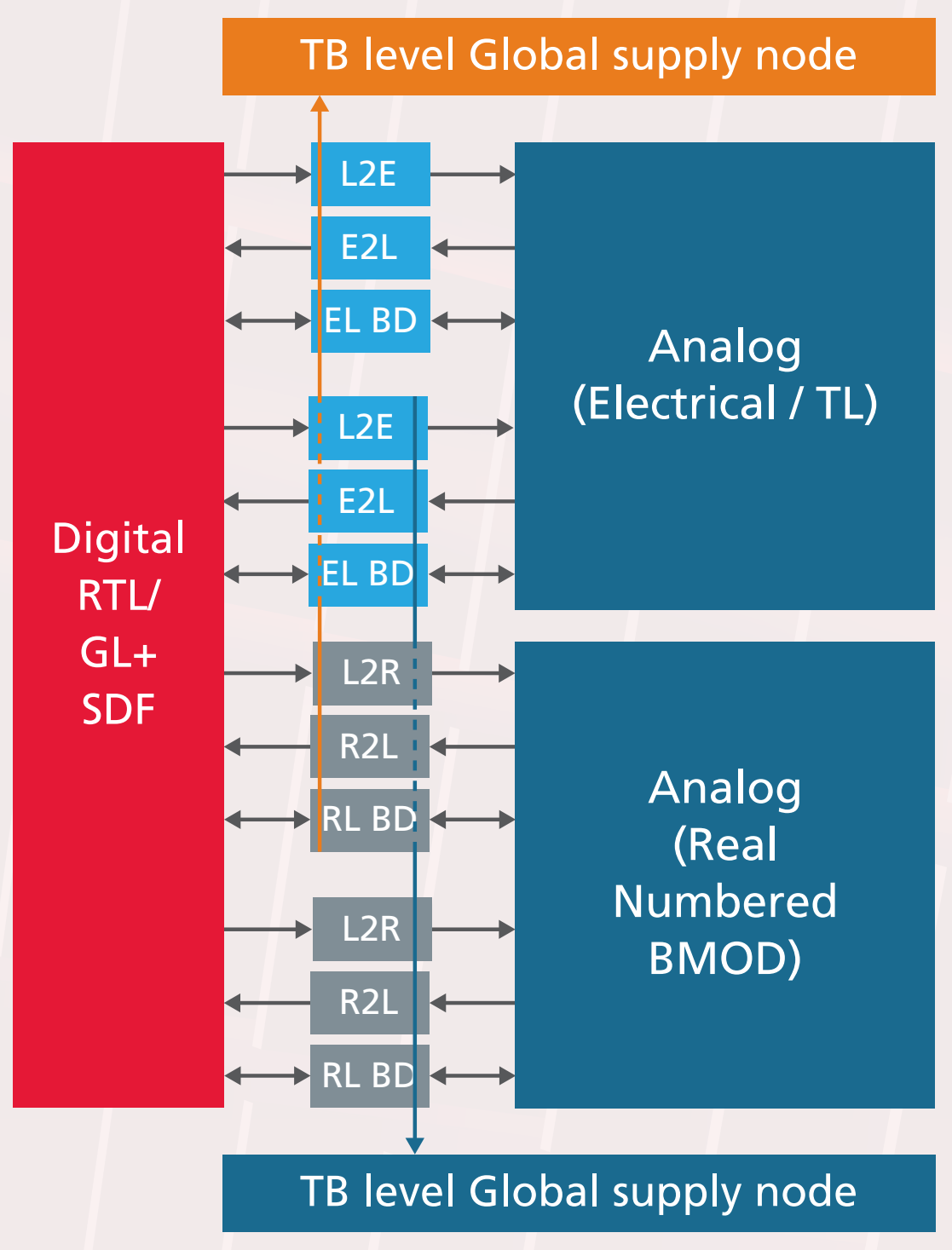


Supply inherited connect modules

- Inherits the top-level power net
  ➔ Uses the voltage value for conversion
- **Definition (Global!) in connect modules overrides physical supply connectivity**

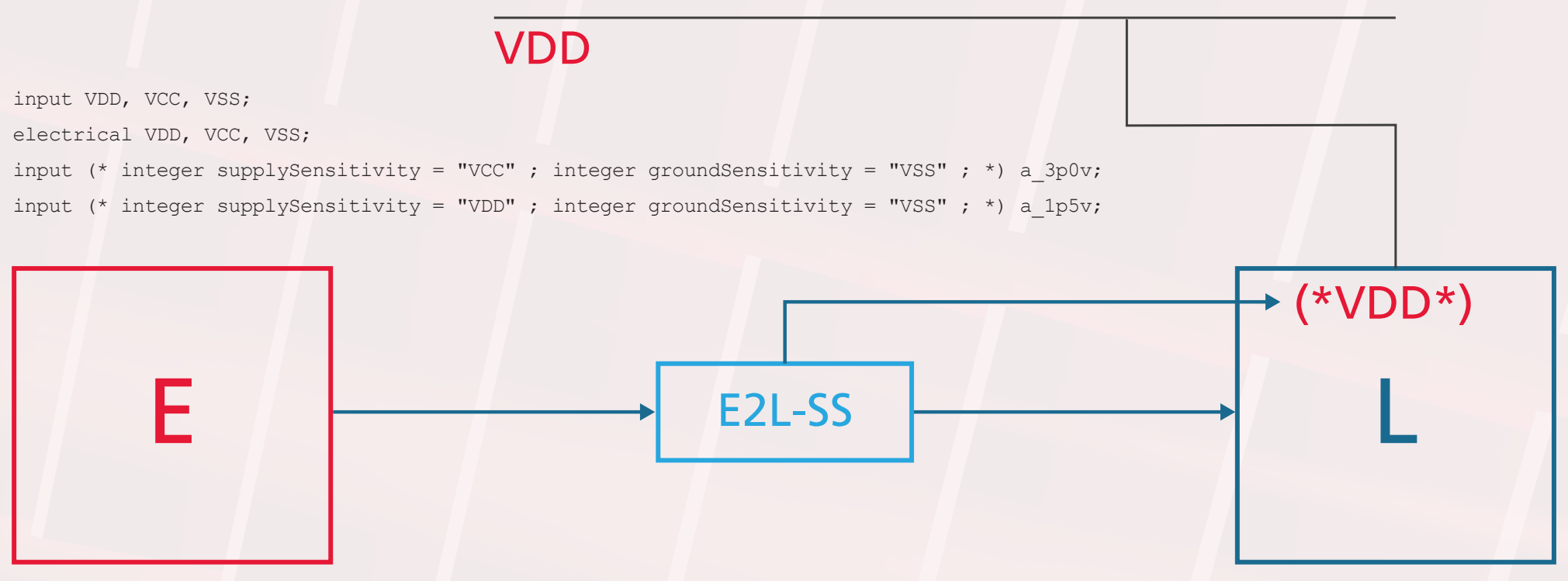
```
 electrical (* integer inh_conn_prop_name="vdd";
     integer inh_conn_def_value="cds_globals.\\vdd!
"; *) \vdd! ;

electrical (* integer inh_conn_prop_name="vss";
     integer inh_conn_def_value="cds_globals.\\vss!
"; *) \vss! ;
```

- Semi-automated voltage domain handling
  - Discipline definitions
    - » Logical: Multiple supply domains
  - **Superficial supply dependence definition (as it is not through physical supply connectivity)**
  - Dynamism supported
    - » Power-up/down (ramp) and mode transitions



Supply sensitive connect module

- References the logic ordinary module (LOM) of the port it is connected to ➔ Uses the supply sensitivity (SS) info of the port
- Requires SS information in all design elements
  - Definitions of standard cells, I/Os, and analog behavioral models



```
input VDD, VCC, VSS;
electrical VDD, VCC, VSS;
input (* integer supplySensitivity = "VCC" ; integer groundSensitivity = "VSS" ; *) a_3p0v;
input (* integer supplySensitivity = "VDD" ; integer groundSensitivity = "VSS" ; *) a_1p5v;
```

- Requirement of hardcoding SS information in all design elements

New Verilog-AMS system functions

VERILOG-AMS LRM c2.4.0, section 9.20 - Analog node alias system functions

```
analog_node_alias_system_function::=
   $analog_node_alias ( analog_net_reference, hierarchical_reference_string )
   $analog_port_alias ( analog_net_reference, hierarchical_reference_string )
```

```
//Electrical to logic CM
connectmodule elect_to_logic(el,cm);
input el; output cm;
reg cm;
electrical el, vdd; ddiscrete cm;
parameter string vddname = " "; // Empty string, Set via the CR
analog initial begin
     if($analog_node_alias( vdd, vddname) == 0)
     $error( "Unable to resolve power supply: %s", vddname);
end
always @( cross(V(el) - V(vdd)/2.0, 1))
     cm = 1;
always @(cross(V(el) - V(vdd)(2.0, -1))
     cm = 0;
endmodule
```

```
//Logic to electrical CM
connectmodule logic_to_elect(cm, el);
input cm; output el;
ddiscrete cm;
electrical el, vdd;
parameter string vddname = " "; // Empty string, set via the CR
     analog initial begin
     if($analog_node_alias(vdd, vddname) == 0)
     $error(”Unable to resolve power supply: %s”, vddname);
     end
analog V(el)<+ V(vdd)*transition((cm == 1) ? 1 : 0);
endmodule
//Connect rules file that instantiates the CMs
connectrules mixedsignal;
connect elect_to_logic#(.vddname("global_supply.vdd”));
connect logic_to_elect#(.vddname("global_supply.vdd”));
endconnectrules
```

Use model of DVS connect modules

- Dynamic voltage supply (DVS) connect modules
- Similar to static connect modules using the Cadence® Incisive® AMS use model flow
  - Uses amsd { } block and ie cards
- Constant vsup replaced with string parameters
  - vddnet/vssnet used for supply/ground specification

```
amsd{
ie connrules = CR_dynsup_full_fast   vddnet="testbench.ams_avdd"
vssnet="testbench.ams_VSS" cell="INV_F1_3P3V “
ie connrules = CR_dynsup_full_fast   vddnet="testbench.ams_vddc"
vssnet="testbench.ams_VSS" cell="INV_F1_HVT “
}
```

Case 1: E2L and L2E DVS connect modules using an electrical supply

Path.Name	Cursor
u_duv.in_1p5v__E2L_2_dynsup__ddiscrete_dynsup_full_fast_ie2.vddnet	"testbench.ams_vddc"
u_duv.in_1p5v__E2L_2_dynsup__ddiscrete_dynsup_full_fast_ie2.vdd_value	1.5
u_duv.in_1p5v__E2L_2_dynsup__ddiscrete_dynsup_full_fast_ie2.Ain	3.3 V
u_duv.in_1p5v__E2L_2_dynsup__ddiscrete_dynsup_full_fast_ie2.Dout	1

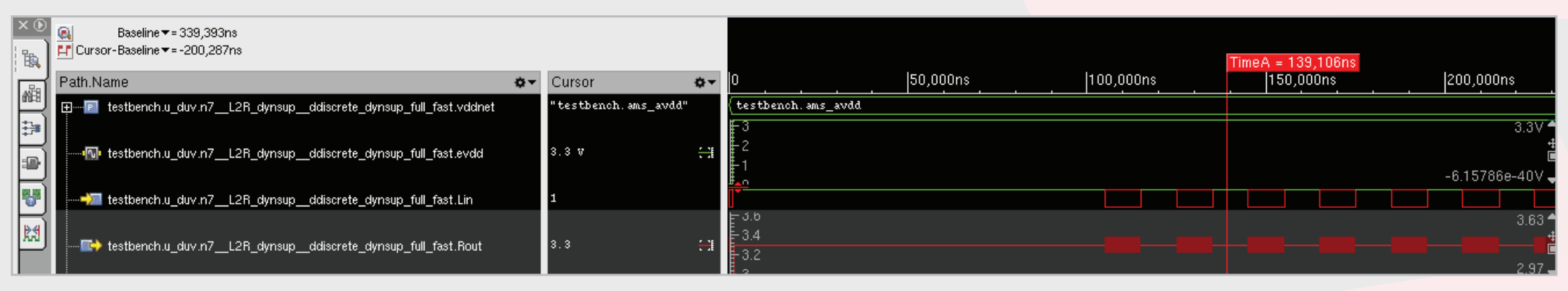
Baseline = 339,393ns
Cursor-Baseline = -169,230ns
TimeA = 170,163ns

Path.Name	Cursor
bench.u_duv.n2__L2E_2_dynsup__ddiscrete_dynsup_full_fast_ie2.vddnet	"testbench.ams_vddc"
bench.u_duv.n2__L2E_2_dynsup__ddiscrete_dynsup_full_fast_ie2.vdd_value	1.5
bench.u_duv.n2__L2E_2_dynsup__ddiscrete_dynsup_full_fast_ie2.Din	1
bench.u_duv.n2__L2E_2_dynsup__ddiscrete_dynsup_full_fast_ie2.Aout	1.5 V

Baseline = 339,393ns
Cursor-Baseline = -158,042ns
TimeA = 181,351ns

Case 2: Heterogeneous DVS connect modules

- R2L/L2R connect modules ideally use wreal supplies
- Heterogeneity feature: Real number to logic conversion based on an electrical supply
  - Useful in designs where all the supplies are electrical



Path.Name	Cursor
testbench.u_duv.n7__L2R_dynsup__ddiscrete_dynsup_full_fast.vddnet	"testbench.ams_avdd"
testbench.u_duv.n7__L2R_dynsup__ddiscrete_dynsup_full_fast.evdd	3.3 V
testbench.u_duv.n7__L2R_dynsup__ddiscrete_dynsup_full_fast.Lin	1
testbench.u_duv.n7__L2R_dynsup__ddiscrete_dynsup_full_fast.Rout	3.3

Baseline = 339,393ns
Cursor-Baseline = -200,287ns
TimeA = 139,106ns

DVS connect modules: Limitations

- Effort needed to set up various ie cards for a power-managed design, if there are tens of power domains in the design
  - A rare scenario, as today's SoCs usually consist of 3 or 4 power domains
- Insertion of connect modules might occur anywhere in the design
  - Decided by the tool's discipline resolution (DR) process
  - Connect module can get placed in an undesired location (can be taken care of)
- DVS connect modules make use of supply information from the verification engineer
  - Contrary to SS connect modules, which make use of the supply information from the IP owner

Conclusion

- Mainly targets verification engineers predominantly using the Incisive® AMS use model flow for digital-on-top designs
  - Incisive AMS use model flow also covers analog-on-top scenarios
- Easy-to-use string parameter constructs that specify the reference hierarchical supply and ground nets
  - Compatible with amsd { } block and 'ie' cards
  - Can easily migrate from inherited connect modules to DVS connect modules
  - Automation scripts if needed
- Multi-power domain designs
  - One-time effort of classifying the cells' and instances' voltage domains
  - Reusable across test cases for coverage
- Design with static connect modules becomes a power-managed design
  - Static voltage thresholds can be simply converted to string parameters