TEXAS INSTRUMENTS

2016
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES

cadence®

A Holistic Approach to Low-Power, Mixed-Signal Design Verification Using Power Intent

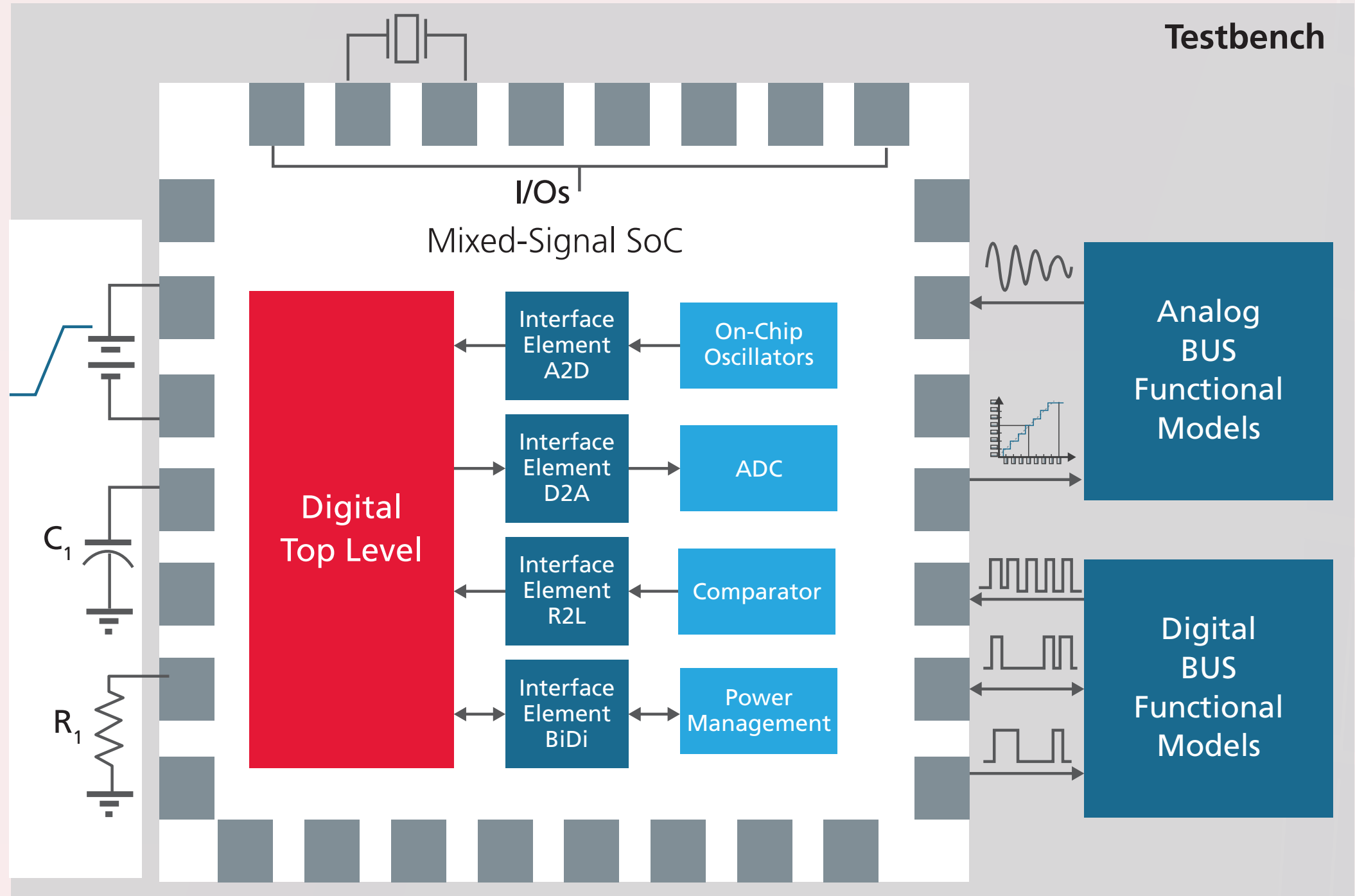
CPF-Based Interface Elements, Methods, and Guidelines

Qingyu Lin, Principal Product Engineer, Cadence

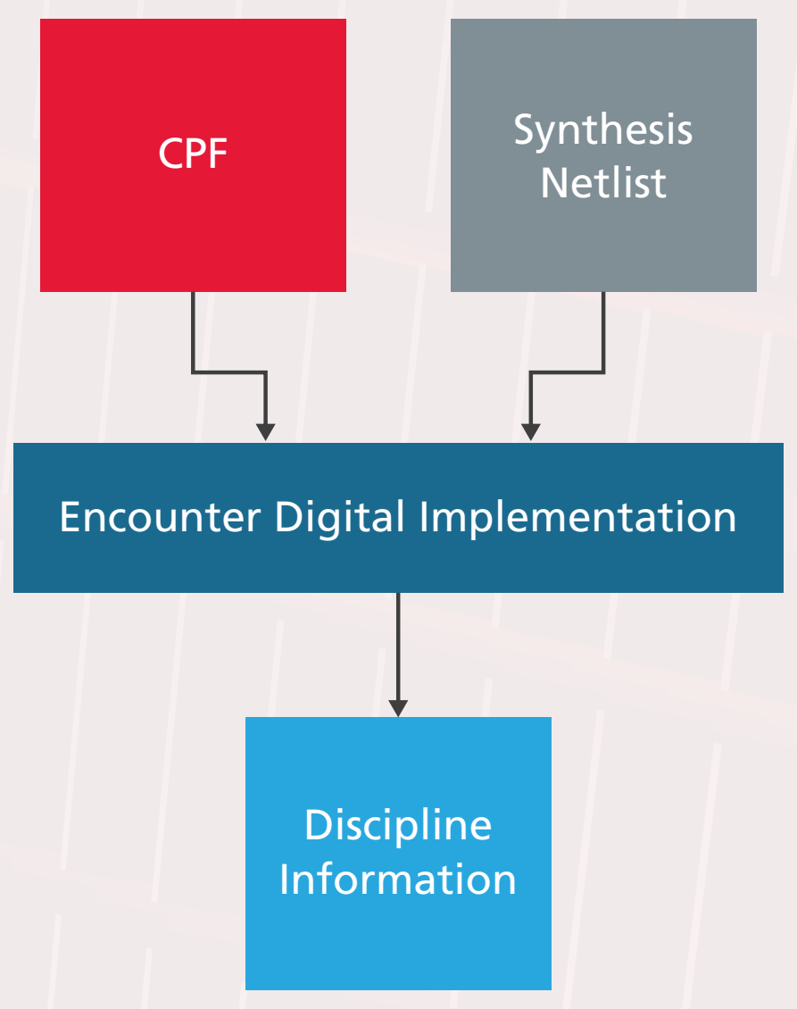
Introduction

- Analog and mixed-signal (AMS) co-simulation ensures device integration and verification quality
- Real number models improve simulation accuracy in digital and mixed-signal (DMS) co-simulation
- AMS and DMS co-simulations inherently have electrical and logical (boolean, real) disciplines
- Tools support interconnection between these disciplines through interface elements (IE), which are of different types selected based on purpose
- Power-managed design needs **supply inherited** or **supply sensitive IEs** to track supply dynamics
- Common Power Format (CPF), **to infer supply sensitivity information** for supply inherited IEs in a multi-voltage/-power domain SoC
- Complexity scales up with **multiple power domains** due to lot of manual effort
- Proposed method to use CPF directly for implementing supply sensitivity in **AMS-RTL** simulations

Mixed-Signal Simulation Environment



Prior Work



CPF

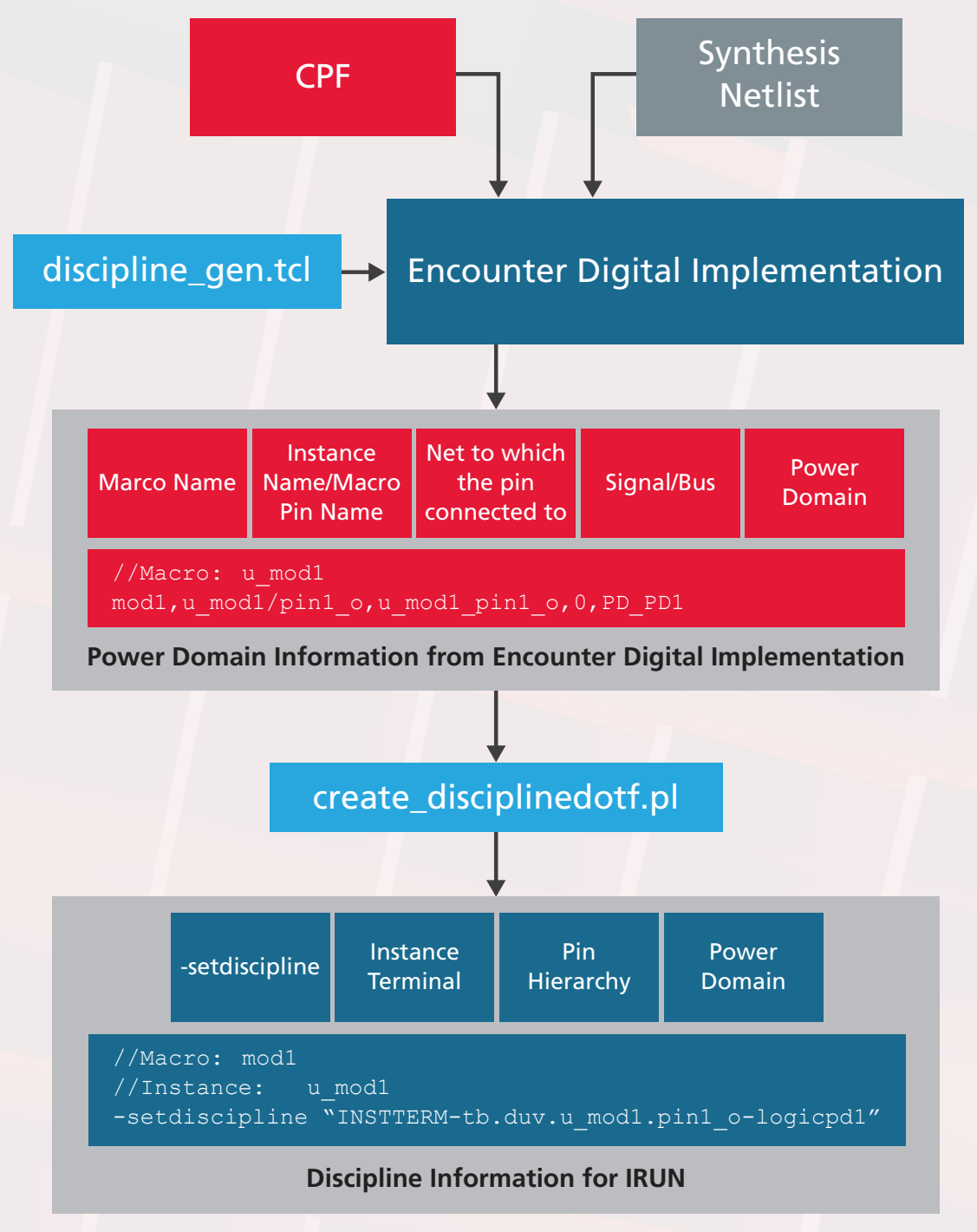
Synthesis Netlist

Encounter Digital Implementation

Discipline Information

- Power intent captured in the **CPF** considered golden for the given **IP** or **SoC**
- This information is sufficient enough to know voltage/power domain of each pin of the **IP**
- **Cadence® Encounter® Digital Implementation** used to get the pin power domains from CPF for all the design elements in the SoC
- Post processed the domain details obtained from **Encounter Digital Implementation** to get discipline information that the simulation tool can understand

Prior Work



CPF

Synthesis Netlist

discipline_gen.tcl

Encounter Digital Implementation

Marco Name	Instance Name/Macro Pin Name	Net to which the pin connected to	Signal/Bus	Power Domain

```
//Macro:  u_mod1
mod1,u_mod1/pin1_o,u_mod1_pin1_o,0,PD_PD1
```

Power Domain Information from Encounter Digital Implementation

create_disciplinedotf.pl

-setdiscipline	Instance Terminal	Pin Hierarchy	Power Domain

```
//Macro:  mod1
//Instance:   u_mod1
-setdiscipline “INSTTERM-tb.duv.u_mod1.pin1_o-logicpd1”
```

Discipline Information for IRUN

```
discipline logicpd1
        domain discrete;
enddiscipline


electrical
    (*integer inh_conn_prop_name=”pd1”;
    integer inh_conn_def_value=”tb.ams.pd1”;*)\pd1!;
electrical
    (*integer inh_conn_prop_name=”gnd”;
    integer inh_conn_def_value=”tb.ams.gnd”;*)\gnd!;


irun -f discipline.f <other_switched>
    -amsconnrules CR_inhconn_full_pd1   connect_pd1/*.vans
    -amsconnrules CR_inhconn_full_pd2   connect_pd2/*.vans
    -amsconnrules CR_inhconn_full_pd3   connect_pd3/*.vans
    -amsconnrules CR_inhconn_full_pd4   connect_pd4/*.vans
```

IE Groups	Connect Rule	IEs
Connect_pd1	CR_full_pd1	E2L_full_pd1
Connect_pd2	CR_full_pd2	E2R_full_pd2
Connect_pd3	CR_full_pd3	R2L_full_pd3
Connect_pd4	CR_full_pd4	L2R_full_pd4

Proposed solution (1/4)

Use a **Verilog-AMS** test bench and a **Verilog-AMS DUT** to establish the physical electrical supply connections at the DUT level

- CPF does not support supply port connection, works only based on the ports’ power domains
- Physical supply connection needed only at the level where electrical and logical design partitions interact directly

```
amodule DUT (P1, P2, P3, VDD1, VDD2, VSS1, VSS2,  F1, F2, F3);
inout VDD1, VDD2, VSS1, VSS2;
input P1, P2, P3;
output F1, F2, F3;
electrical VDD1, VDD2, VSS1, VSS2;
…
endmodule
```

Proposed solution (2/4)

Define the **power intent** for the SoC in **CPF**, constructs for illustration below

```
acreate_power_domain -name PD_1 –default
update_power_domain -name  PD_2 -primary_power_net VDD1 -prima-
ry_ground_net VSS1
set_instance u_inst_1 -model model1 -domain_mapping { {PD_VDD1_VSS1
PD_DIG_1 }  {PD_VDD2_VSS2 PD_VDD3 } }
create_state_retention_rule -name RET1 -instances    { tb/du-
v/u_dig/u_ip_wrap/u_0 tb/duv/u_dig/u_ip_wrap/u_1 } \
-save_edge  {tb/shutoff_condit  ion } -restore_edge  {!tb/shutoff_condi-
tion }
set_macro_model flash_ ip
create_nominal_condition -name N1 -voltage 3.3 -ground_voltage 0
create_nominal_condition -name OFF -voltage 0 -ground_voltage 0 -state off
create_power_domain -name PD_VDD2_flash -boundary_ports {flash_3P3V
clk_3P3V en_flash_3P3V ……}
update_power_domain -name PD_VDD2_flash -primary_power_net VDD_flash
-prima  ry_ground_net VSS  -equivalent_power_nets { VDD2 VDD3 VDD4 }
-equivalent_ground_nets { VSS1 VSS2 VSS2 VSS4 }
create_power_mode -name PM1 -default -domain_conditions {PD_VDD2_flash@N1
}
create_power_mode -name PM2 -domain_conditions { PD_VDD2_flash@OFF }
end_macro_model
```

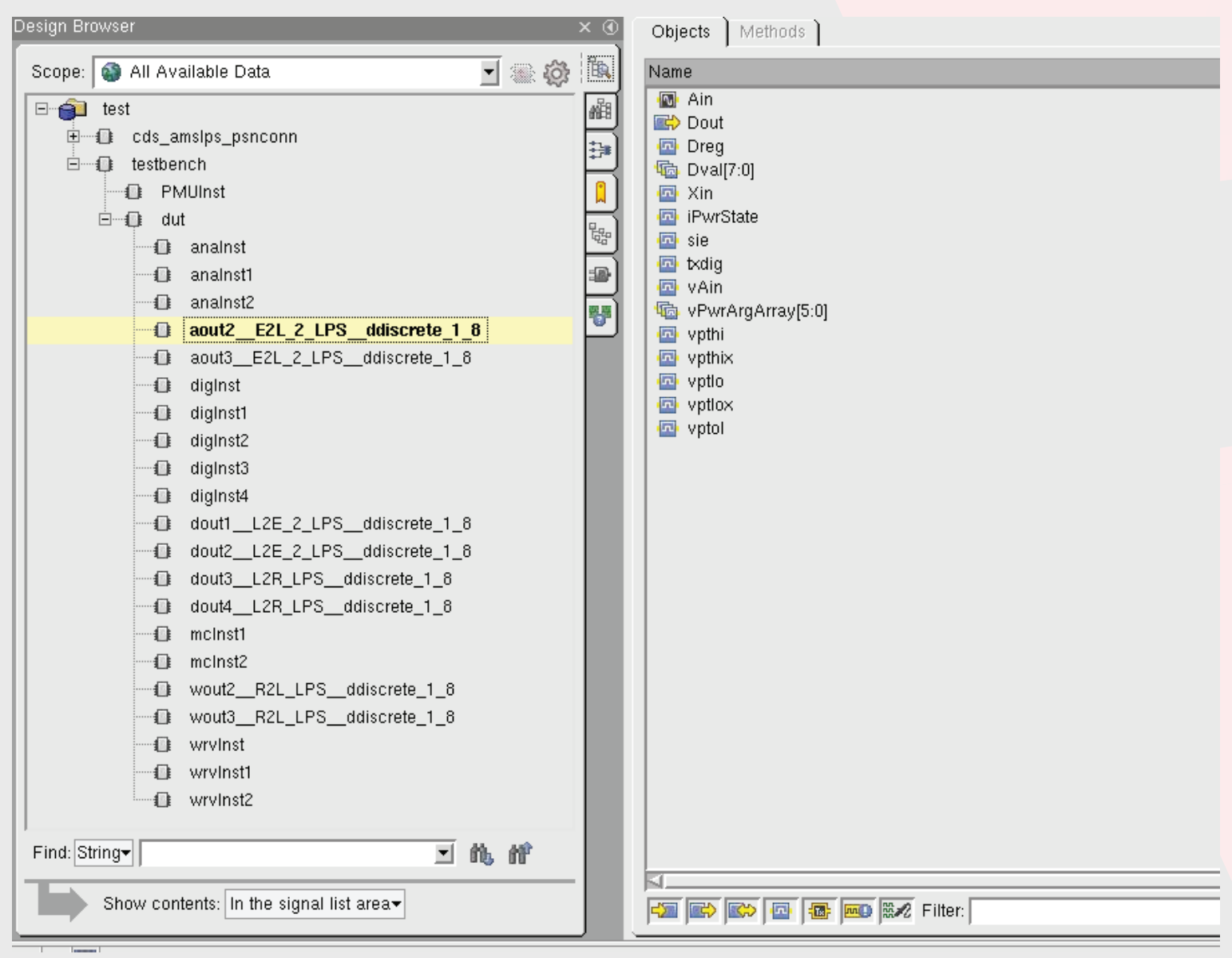
Proposed solution (3/4)

Simulate the design with **CPF** by including these commands in **IRUN**:

- **-amsconnrules CR_full_fast** to use Cadence provided built-in power-smart CPF-based IE
- **-discipline logic** to set a global logic discipline to the entire design
- **-ams_generate** to support IEs inside VHDL generate statements
- **-lps_pmode** to enable power mode simulation

Proposed solution (4/4)

- Check the CPF or power-smart IE insertion by referring the tool-generated IE info file and verify the design functionality
  - CPF-based IEs have __**LPS** suffix to denote **l**ow **p**ower **s**imulation



Results

- Use of CPF directly for IE insertion in DMS and AMS is most efficient and accurate
- Completely eliminated false failures due to manual errors which is very costly for AMS co-simulations as these simulations require very long run times
- Completely eliminated the overhead of deriving discipline information for the given SoC, unlike supply inherited IEs that required a lot of manual work
- The flow is very generic and can be implemented with almost zero effort in any power-managed SoC

Limitations and future scope

- CPF-based IEs can be used only for RTL AMS co-simulations, but not for gate-level (GL) AMS co-simulations
  - Physical power ports connectivity in GL netlist will be overridden by CPF
  - Need supply sensitivity based on physical connection to support such requirements
- Need for physical supply port connections in CPF to track the supply variations
  - CPF constructs do not support supply port connection, works only based on the power domains of the ports
  - Supply variation (fluctuation) cannot be tracked by CPF
  - Need to make the physical electrical supply connection from the DUT to the IPs using Verilog-AMS test bench and DUT
  - Work under progress by Cadence to enable supply port connection in CPF as a proprietary feature