# Unifying Mixed-Signal and Low-Power Verification

Adam Sherer, Andre Baguenier, Kawe Fotouhi,
Abhijit Madhu Kumar, Qingyu Lin, Raj Mitra, William Winkeler

cadence®

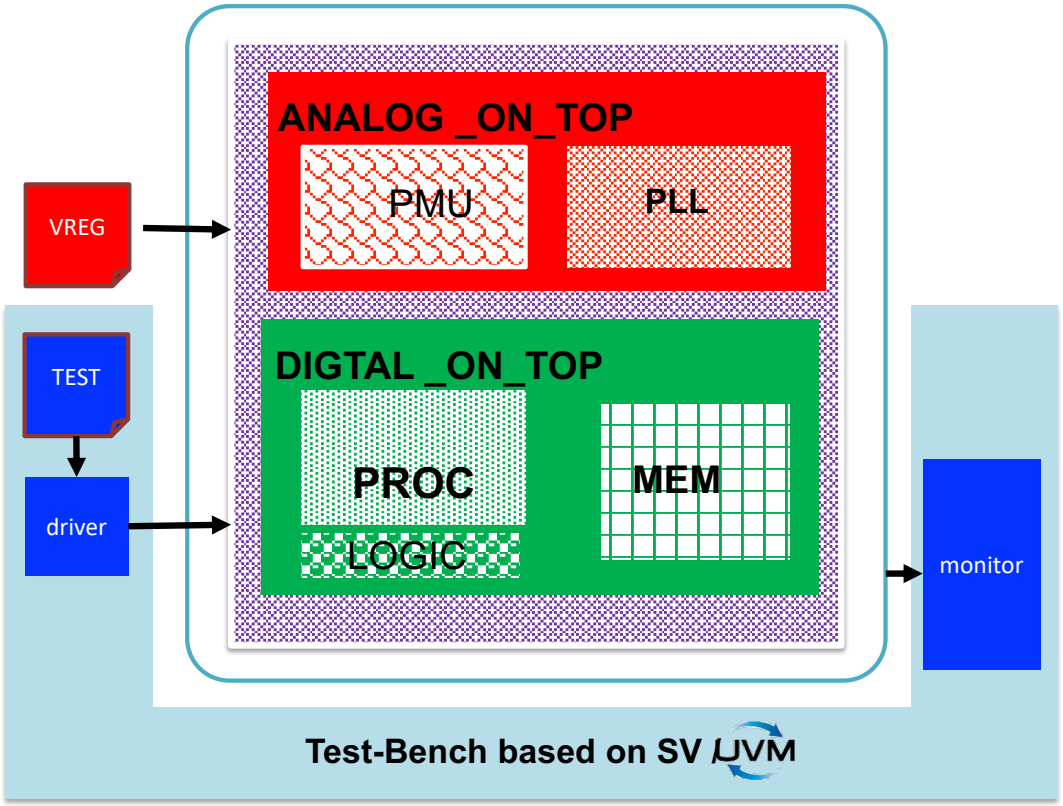
accellera SYSTEMS INITIATIVE

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

# Agenda

- Introduction
  - SoC Example architecture overview
  - Selection of flow
  - Analog core overview (Andre)
  - Digital core overview (Kawe)
- Digital Core Power Intent (Kawe)
  - LP1801 introduction
  - Creation
  - Verification (demonstration)
- SoC Power Supply  Creation (Andre)
  - Introduction
  - Step by step creation with real time demonstration
- Modeling Dynamic  Power Loading (Andre)
  - Modeling loading effect in RNM
  - EEnet intro
  - Demonstration
- Virtuoso IP Bloc Export and Reuse (Andre)
  - Command line IP CLIPS introduction
  - Demonstration
- Using UVM-MS (Kawe)
  - UVM MS intro
  - UVM demonstration
- Conclusion and wrap-up

# SoC Overview




**MCU with off-chip regulator :**

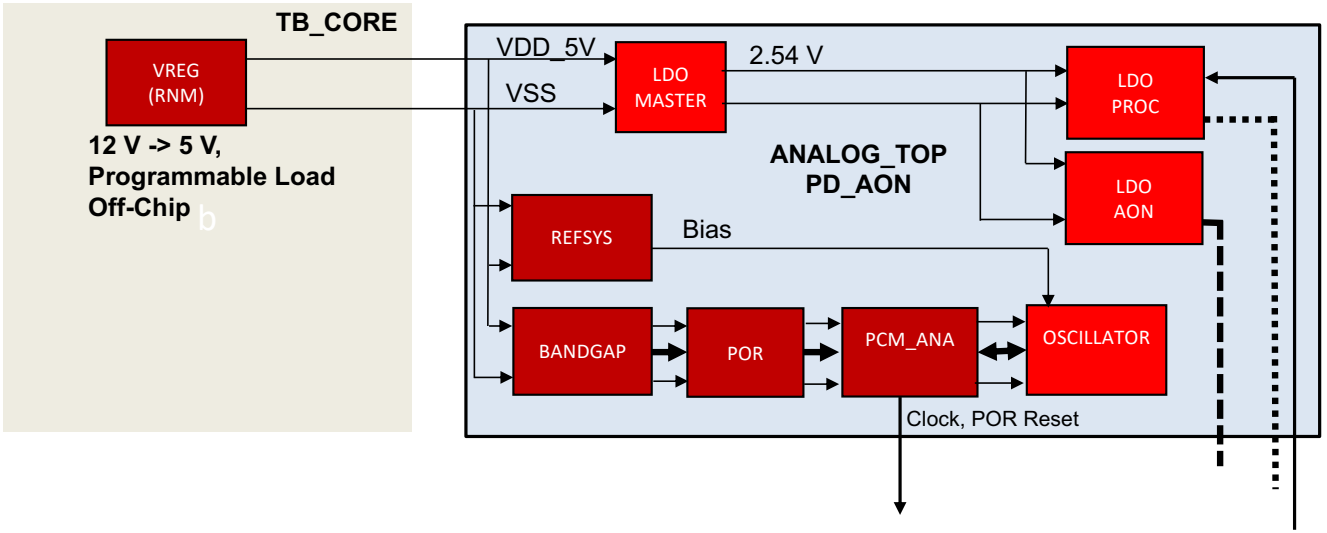
- on-chip supplies PMU
- Three LDOs
- PLL

analog

- Processor
- Mem
- Cache
- SPI
- SRAM
- Power control

digital

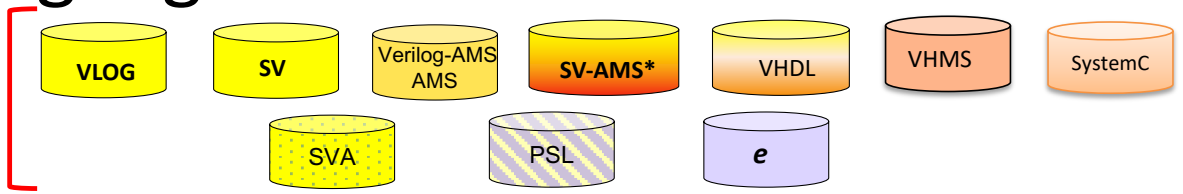
# Analog Top Block Diagram & Main Specifications




| Features | Specification |
|---|---|
| F. CLOCK | 160MHz ± 10MHz |
| PLL CLOCK Lock | <=35us |
| VDD_AON | 1.25V ± 0.2V |
| VDD_PROC | 0.9V ± 0.1V |

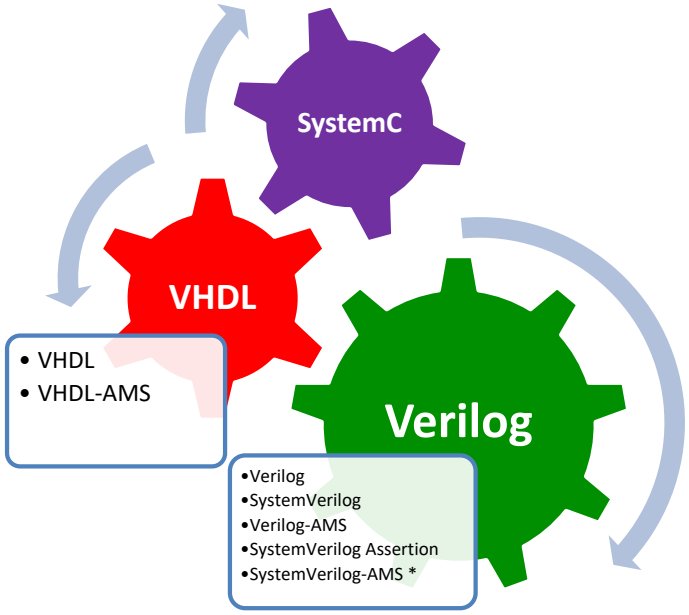
# Flow Selection
# Which Languages Should We Use?

Xcelium™ can compile




* Today Limited to connect module

Is it a good idea to mix all languages in TB and DUT?




- It is possible to mix all but …

Sandwich or lasagna recipes are not a good idea. (signals propagation, coercion, DR, AICMS,….)

Prefer isolated island when mixing languages

accellera SYSTEMS INITIATIVE

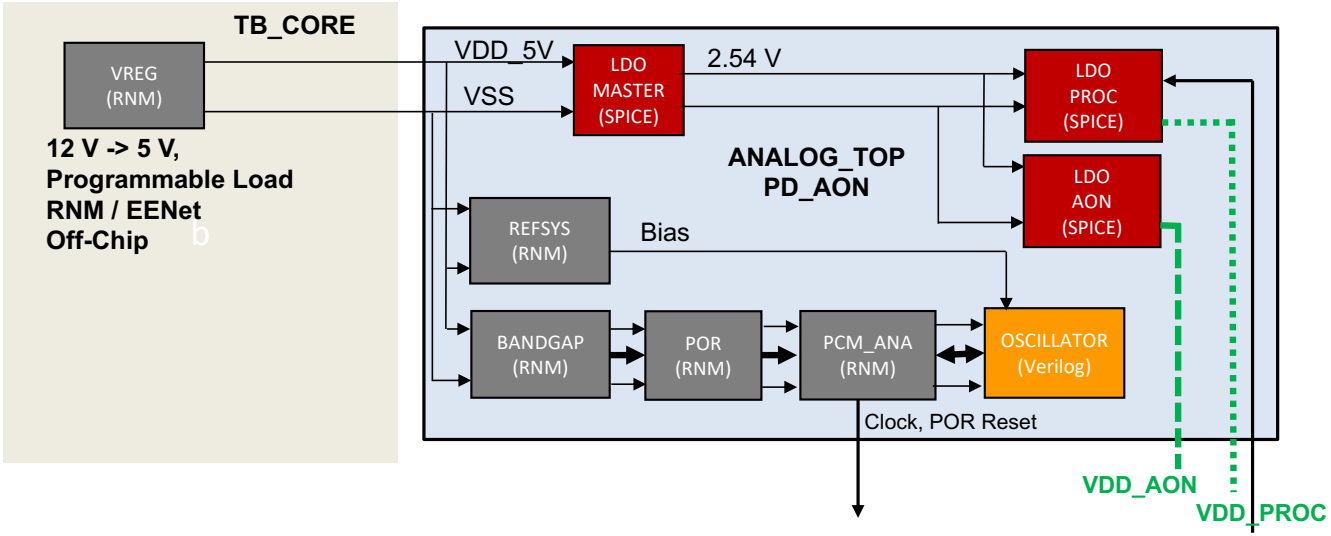


2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

# Real-Number Model (RNM): Which language?

| Features | VHDL RNM | Verilog-AMS RNM | SystemVerilog RNM |
|---|---|---|---|
| real signal | Y ✅ | Y wreal ✅ | Y real, nettype ✅ |
| default resolution function | N ⛔ | Y 5 build-in ✅ | No, but use cds_rnm_pkg |
| custom UDT | Y ✅ | N ⛔ | Y ✅ |
| custom UDR | Y ✅ | N ⛔ | Y ✅ |
| custom UDN | Y ✅ | N ⛔ | Y ✅ |
| wire to wreal coercion 😀 | N ⛔ | Y ✅ | Y ✅ |
| Real2Electrical 😀 | N ⛔ | Y ✅ | Y ✅ |
| Real2logic 😀 | N ⛔ | Y ✅ | Y ✅ |
| Compile order dependency | Y (issue when IP reuse) | N ⛔ | N ✅ |
| Assertion 😀 | Y ✅ | Y ✅ | Y ✅ |
| Bind 😀 | N ⛔ | N ⛔ | Y ✅ |
| Coverage 😀 | N ⛔ | Y ✅ | Y ✅ |
| Randomization 😀 | N ⛔ | N ⛔ | Y ✅ |
| Low power CM 😀 | N ⛔ | Y ✅ | Y (wreal) ✅ |
| Import package from other language | N ⛔ | N ⛔ | Y ✅ with Xcelium™ |

# Analog Top Block Diagram & Flow Choices




| Features | Specification |
|---|---|
| F. CLOCK | 160MHz ± 10MHz |
| CLOCK Lock | <=45us |
| VDD_AON | 1.25V ± 0.2V |
| VDD_PROC | 0.9V ± 0.1V |

SPICE

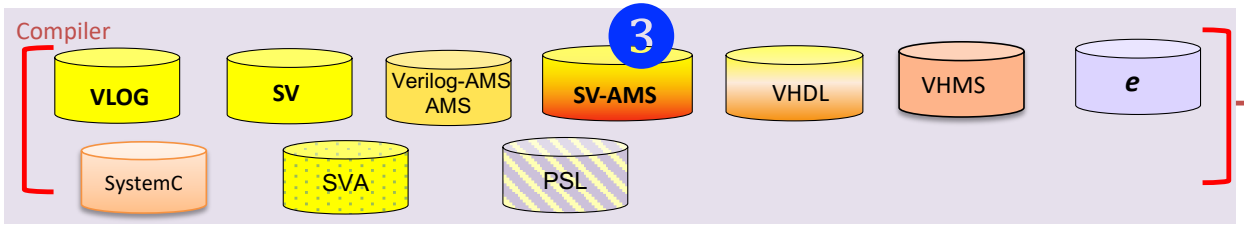
RNM

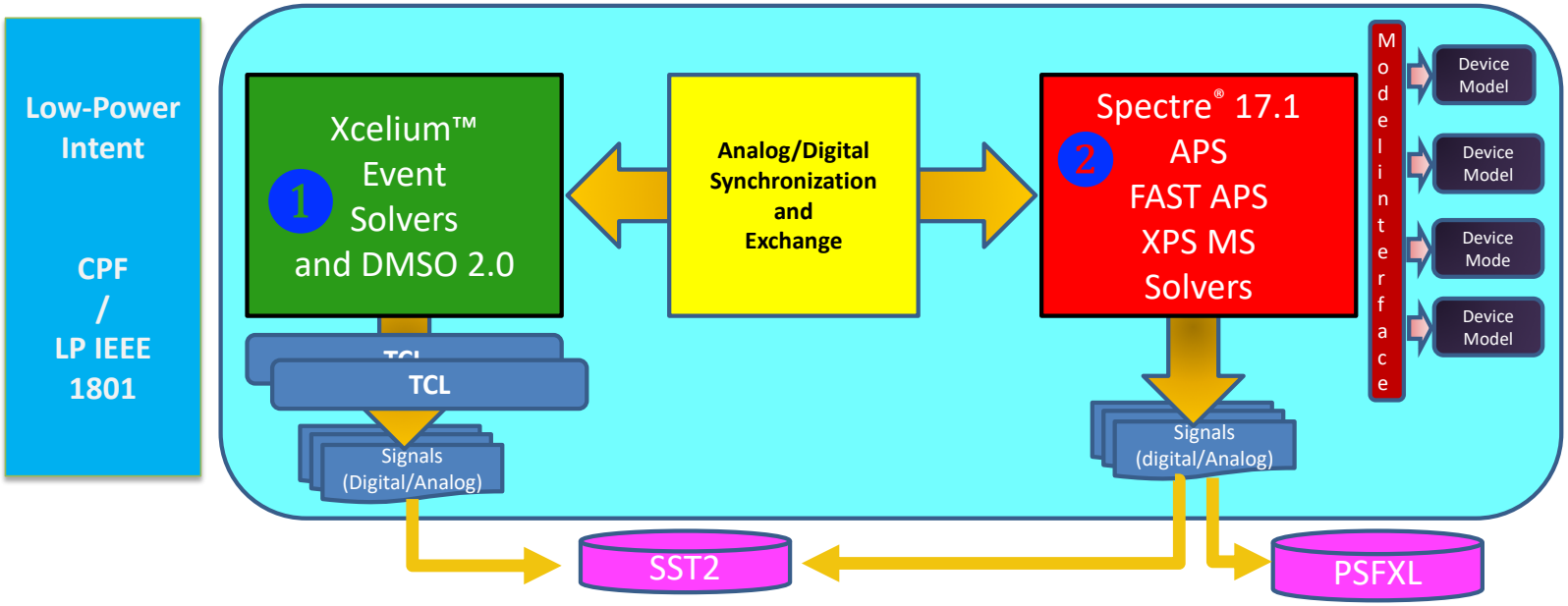
UPF

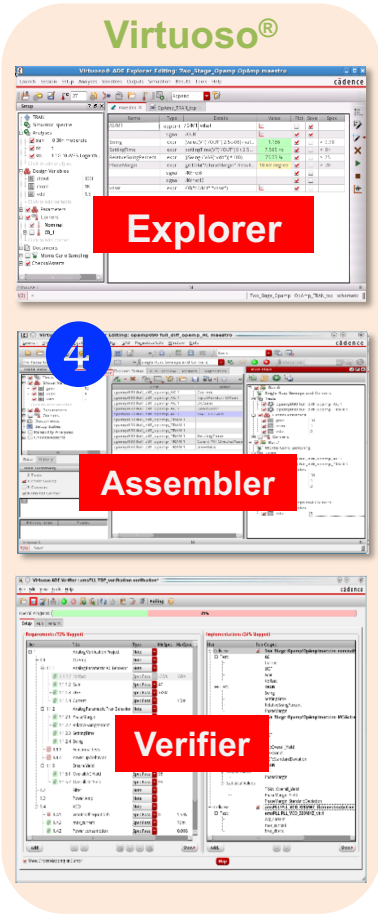
Verilog / SV

UPF Supply Net

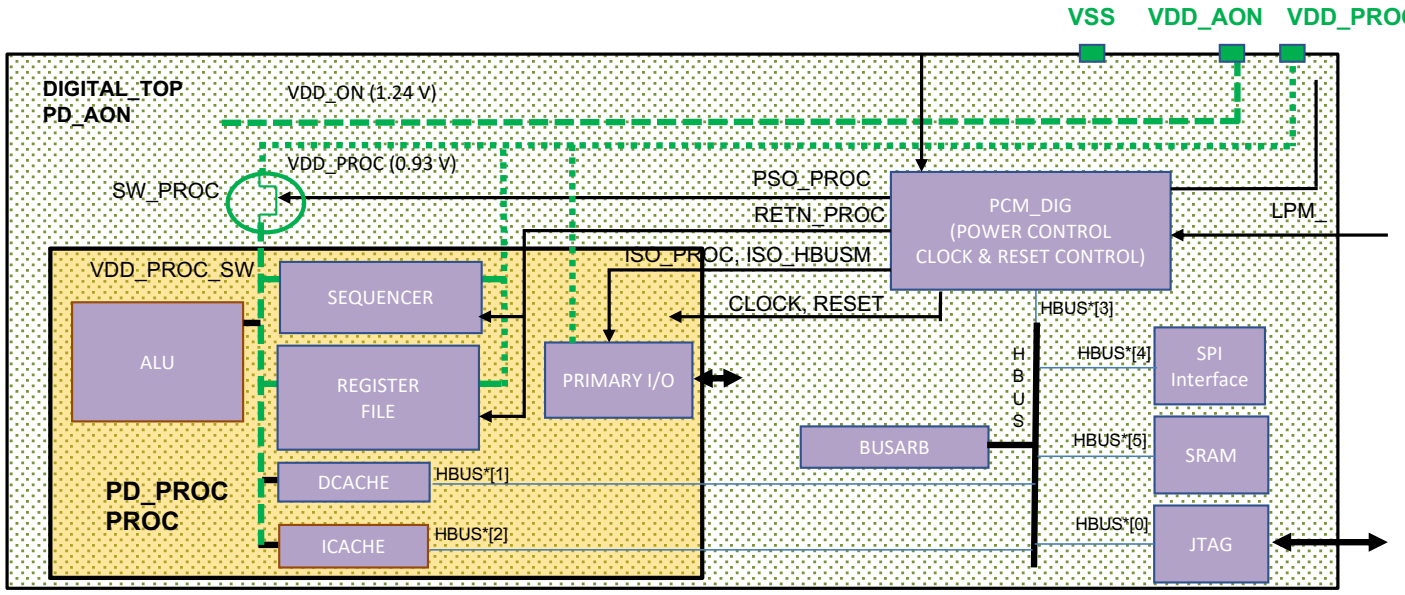
# 4 Main Reasons Explaining Why using Cadence MS Flow



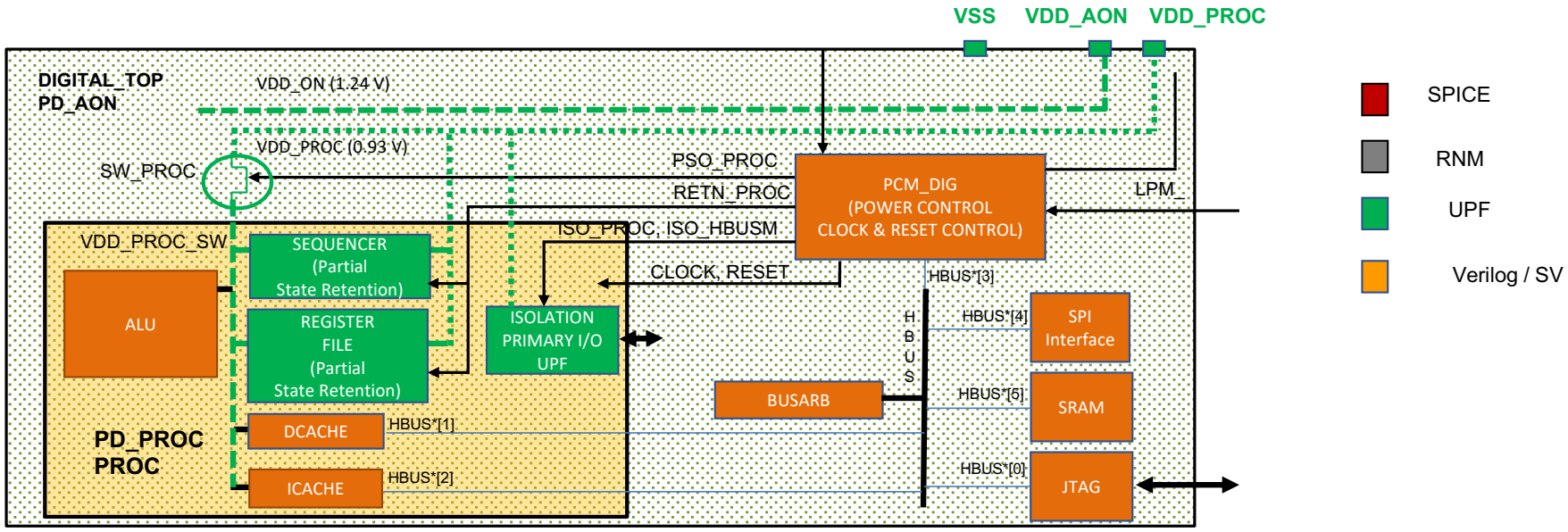



# Digital Top Block Diagram & Main Specifications

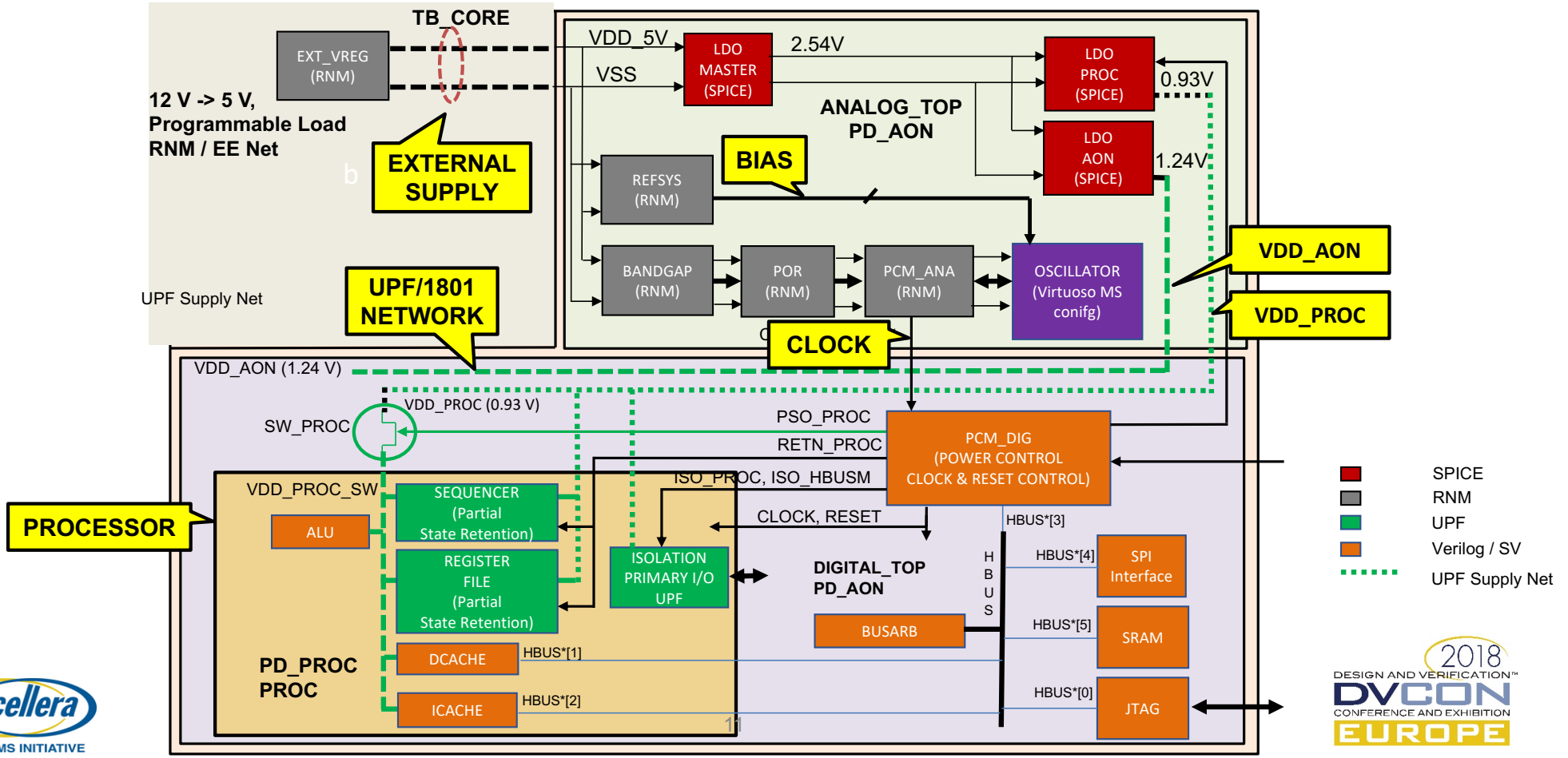



| Features | Specification |
|---|---|
| MCU | RISC / 160Mhz |
| Memory | SRAM 4KB |
| PCM | 1.2V / AON / |
| - clock /reset | POR and LPM |
| HBUS | 160Mhz |
| - JTAG | 25Mhz |

# Digital Top Block: Implementation



# Mixed Signal Design Architecture and Configuration



# What need to be verified?

- Correctness of power network architecture
  - Are the correct instances in the intended power domains?
  - Are the isolation and retention objects complete and correct?
  - Power-supply, -net, -switches etc.
- Device functionality in Power context
  - What is going to happen when you shut off the power?
    - Will incorrect blocks of the design get corrupted?
  - What is going to happen when you isolate? Will control lines be isolated incorrectly to an active state and lock up the bus?
  - What is the interaction between multiple domains?
  - What is happening when the supply drop?
  - Does power and reset works?
  - Will the design come back up in the right state as expected (data lost etc.)?
- Power efficiency (not in context of this tutorial)
  - Have you met the power budget?

# Agenda

- Introduction
  - SoC Example architecture overview
  - Selection of flow
  - Analog core overview (Andre)
  - Digital core overview (Kawe)
- Digital Core Power Intent (Kawe)
  - IEEE 1801 introduction
  - Creation
  - Verification (demonstration)
- SoC Power Supply  Creation (Andre)
  - Introduction
  - Step by step creation with real time demonstration
- Modeling Dynamic  Power Loading (Andre)
  - Modeling loading effect in RNM
  - EEnet intro
  - Demonstration
- Virtuoso IP Bloc Export and Reuse (Andre)
  - Command line IP CLIPS introduction
  - Demonstration
- Using UVM-MS (Kawe)
  - UVM MS intro
  - UVM demonstration
- Conclusion and wrap-up

# IEEE 1801 introduction - Basics -

| Concept | Description |
|---|---|
| Power Domains | Group the elements of logic hierarchy that share the same primary power supply |
| Supply Ports | Provide the supply interface to power domains and switches |
| Supply Nets | Connect supply ports |
| Power Switch | Based on the value of the power control signal, the Power Switch connects / disconnects the input supply port to the output supply port of the switch |
| HDL Supply Net Control Functions | UPF provides functions which enable the user to drive Supply Ports in low power simulation:<br>**`supply_on`**, **`supply_off`**, **`supply_partial_on`** |
| Power Supply Network | Consists of supply ports, supply nets and power switches and their interconnections |
| LDO | Low-dropout regulator. DC/DC converter used for on-chip power supplies |

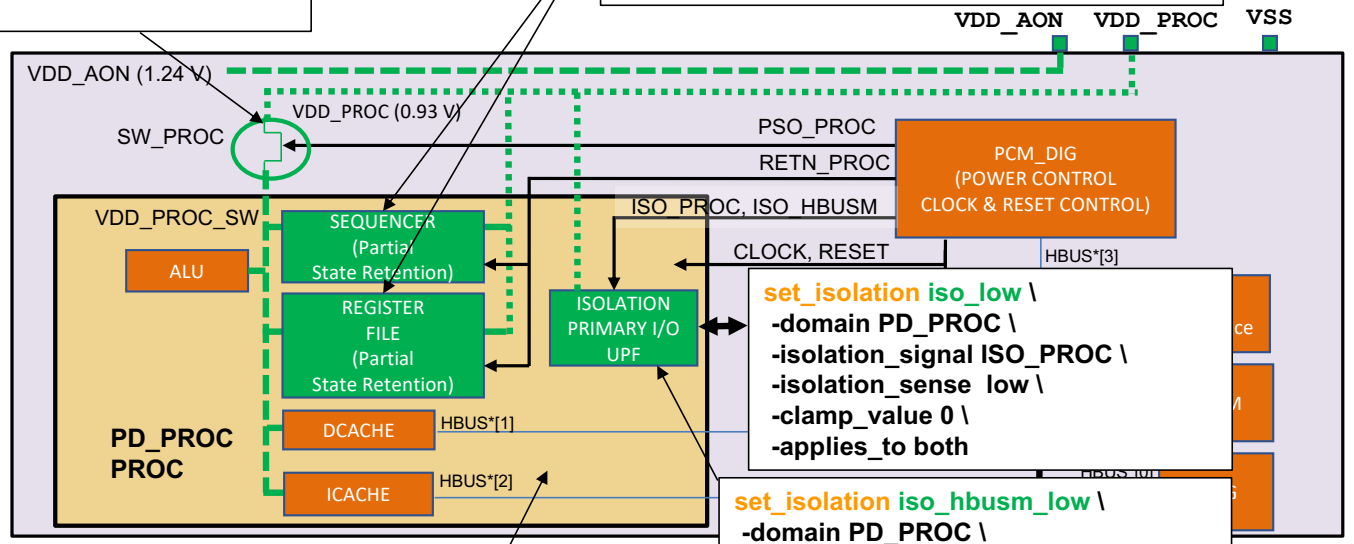
# IEEE 1801 introduction - Basics -

| Concept | Description |
|---|---|
| State Retention | Allows the contents of registers to be saved prior to power shutoff and recovered when is power is restored<br>Usually performed on key control registers |
| Isolation | Prevents corrupted values from propagating from shutoff power domains to power domains which are powered up |
| Power Shutoff (PSO) | Power reduction method where power domains are shutoff. Shutoff can be performed by Power Switch or by turning off the power to the supply ports.<br><br>Isolation and State Retention are often used in Power Shutoff Domains |

# DIGITAL_TOP - PARTIAL UPF -

```
create_power_switch SW_PROC \
    -input_supply_port {VIN VDD_PROC}\
    -output_supply_port {VOUT VDD_PROC_SW} \
    -control_port {EN PSO_PROC} \
    -on_state {state_on VIN {EN}} \
    -off_state {state_off {!EN}}
```

```
set_retention retn_PD_PROC \
-domain PD_PROC \
-save_signal {RETN_PROC low}\
-restore_signal {RETN_PROC high} \
-elements {PROC/REGISTERFILE …
```




```
set_isolation iso_low \
 -domain PD_PROC \
 -isolation_signal ISO_PROC \
 -isolation_sense  low \
 -clamp_value 0 \
 -applies_to both
```

```
set_isolation iso_hbusm_low \
 -domain PD_PROC \
 -isolation_signal ISO_HBUSM \
 -isolation_sense  low \
 -clamp_value 0 \
 -elements {PROC/HBUSM_REQ …
```

```
create_power_domain PD_PROC \
    -supply {primary SS_PSO} \
    -supply {default_isolation SS_PROC} \
    -supply {default_retention SS_PROC}
```

# Power intent/network specified by IEEE 1801 (UPF 2.0)

- Digital block - Processor -
  - Power saving by switching of the supplies
  - Power domains, state retention, port isolation
- Virtual supply network using  supply_net_type (IEEE 1801)
  - requires STATE and VOLTAGE
- Supply sources
  - Static HDL Supply Net Control Functions ($supply_on)
  - LDO (SPICE / wreal) driving UPF Power Supply Network

SW1, SW2 – UPF Power Switches
VDD_SW1 – UPF Supply Net
VDD_SW2 – UPF Supply Net
VDD3 – UPF Supply Net with Resolution Function

# Power States

| Power State | Description |
|---|---|
| Power Up | LDO AON powers up, Power On Reset,<br>Clock Enabled, JTAG, BUSARB, PCM_DIG on |
| Load SRAM | JTAG loads PROC object code<br>LDO_PROC powers up<br>PROC executes instruction thread |
| Power Down PROC | LPM_ asserted. Cache flush started, clock gated, state saved, isolation enabled.<br>Power Shutoff by Power Switch SW_PROC |
| Power Up<br>PROC | LPM_ released. PROC power on – SW_PROC turned on. Restore state, release state. |
| LDO Shutdown | Output of VREG is heavily loaded, causing LDO_AON and LDO_PROC to be shutoff.<br>Load is removed, enter Power Up State.<br>After POR, enter into LOAD SRAM state<br>After LOAD SRAM – PROC executes instruction thread |

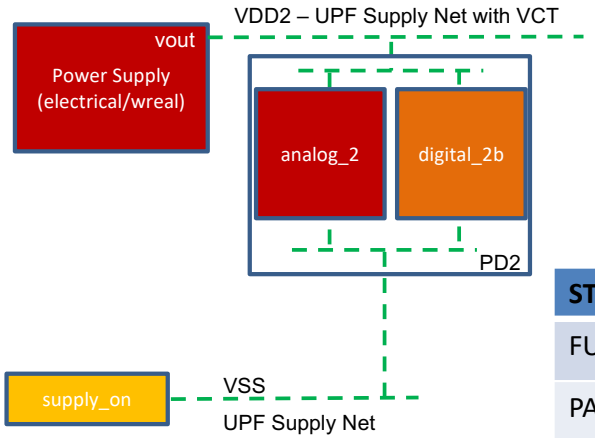
# Demonstration #1

# Agenda

- Introduction
  - SoC Example architecture overview
  - Selection of flow
  - Analog core overview (Andre)
  - Digital core overview (Kawe)
- Digital Core Power Intent (Kawe)
  - IEEE 1801 introduction
  - Creation
  - Power-up verification (demonstration)
  - Power shut-off  verification (demonstration)
- SoC Power Supply  Creation (Andre)
  - Introduction
  - Step by step creation with real time demonstration
- Modeling Dynamic  Power Loading (Andre)
  - Modeling loading effect in RNM
  - EEnet intro
  - Demonstration
- Virtuoso IP Bloc Export and Reuse  (Andre)
  - Command line IP CLIPS introduction
  - Demonstration
- Using UVM-MS (Kawe)
  - UVM MS intro
  - UVM demonstration
- Conclusion and wrap-up

# Driving PSN with electrical / wreal Ports

- UPF Supply Nets require a STATE and VOLTAGE
  - STATE – UNDETERMINED, PARTIAL_ON, FULL_ON, OFF
- wreal / electrical ports provide the VOLTAGE, but no STATE
- Add STATE through VCT (Value Conversion Table)




```
create_supply_net VDD2
create_hdl2upf_vct VCTwr2upf_VDD2 \
-hdl_type {sv cds_rnm} \
-table {{>=4.8 FULL_ON} \
        {>=4.5 PARTIAL_ON} \
        {<4.5 OFF}}
```

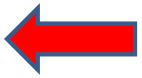
| STATE | Digital Logic |
|---|---|
| FULL_ON | Does not cause corruption |
| PARTIAL_ON | Enable / Disable corruption through UPF command. Default = Corrupt |
| OFF | Corrupt |
| UNDETERMINED | Corrupt |

# Power Supply Network Connectivity Creation with UPF

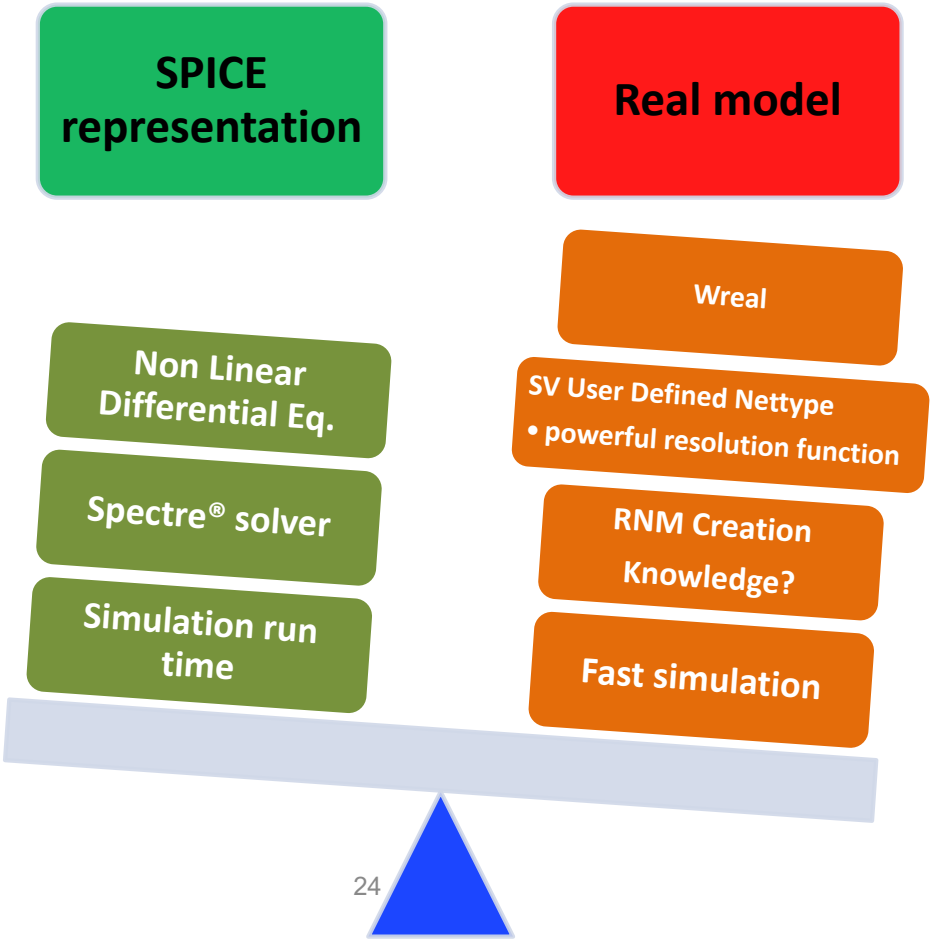
## Demonstration #2

# Agenda

- Introduction
  - SoC Example architecture overview
  - Selection of flow
  - Analog core overview (Andre)
  - Digital core overview (Kawe)
- Digital Core Power Intent (Kawe)
  - IEEE 1801 introduction
  - Creation
  - Power-up verification (demonstration)
  - Power shut-off  verification (demonstration)
- SoC Power Supply  Creation (Andre)
  - Introduction
  - Step by step creation with real time demonstration
- Modeling Dynamic  Power Loading (Andre)
  - Modeling loading effect in RNM
  - EEnet intro
  - Demonstration
- Virtuoso IP Bloc Export and Reuse  (Andre)
  - Command line IP CLIPS introduction
  - Demonstration
- Using UVM-MS (Kawe)
  - UVM MS intro
  - UVM demonstration
- Conclusion and wrap-up



# Modeling Dynamic Power Loading



# EE_pkg.sv Defines Three Important Objects

- A Cadence IP

```
typedef struct {
    real V;
    real I;
    real R;
} EEstruct;
```

UDT (user defined type)

```
function automatic EEstruct res_EE (input EEstruct driver[]);
```

Resolution function

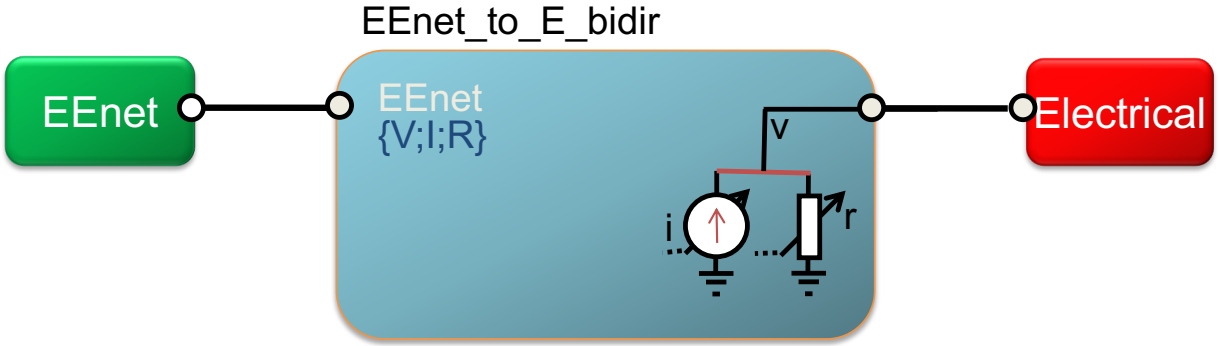
```
// Electrical "EEnet" nettype with thevenin equivalent resolution
// Usage:
//   ideal voltage drive:   V=v1, R=0, (I ignored)
//   ideal current drive:   I=i1, R=`Z and/or V=`Z
//   V+R drive:        V=v1, R=r1, I=0 or `Z
//   I||R drive:           I=i1, R=r1, V=0
//   Combination:             V=v1, R=r1, I=i1
//   No drive:                          I=`Z and (V=`Z or R=`Z)
// Multiple ideal voltage drives will result in `X.
// Ideal currents into large loads or Z will saturate to the
//   specified VHI or VLO voltages.
// Undriven nodes will resolve to the specified VZ,RZ values.
```

```
nettype EEstruct EEnet with res_EE;
```

EEnet UDN (user net type)

# What is an SV-AMS Connect Module (CM) or Interface Element (IE) E_EEnet_Bidir?

- Converter between EEnet and electrical signal.
- In SystemVerilog EEnet is a User Net Type (UDT with resolution function)
- In the electrical solver, it is a Norton generator



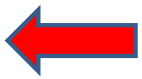

- How do we create a E_EEnet_Bidir?
  - Use the new SystemVerilog-AMS compiler
  - File extension .svams

# Modeling Dynamic Power Loading with RNM

## Demonstration #3

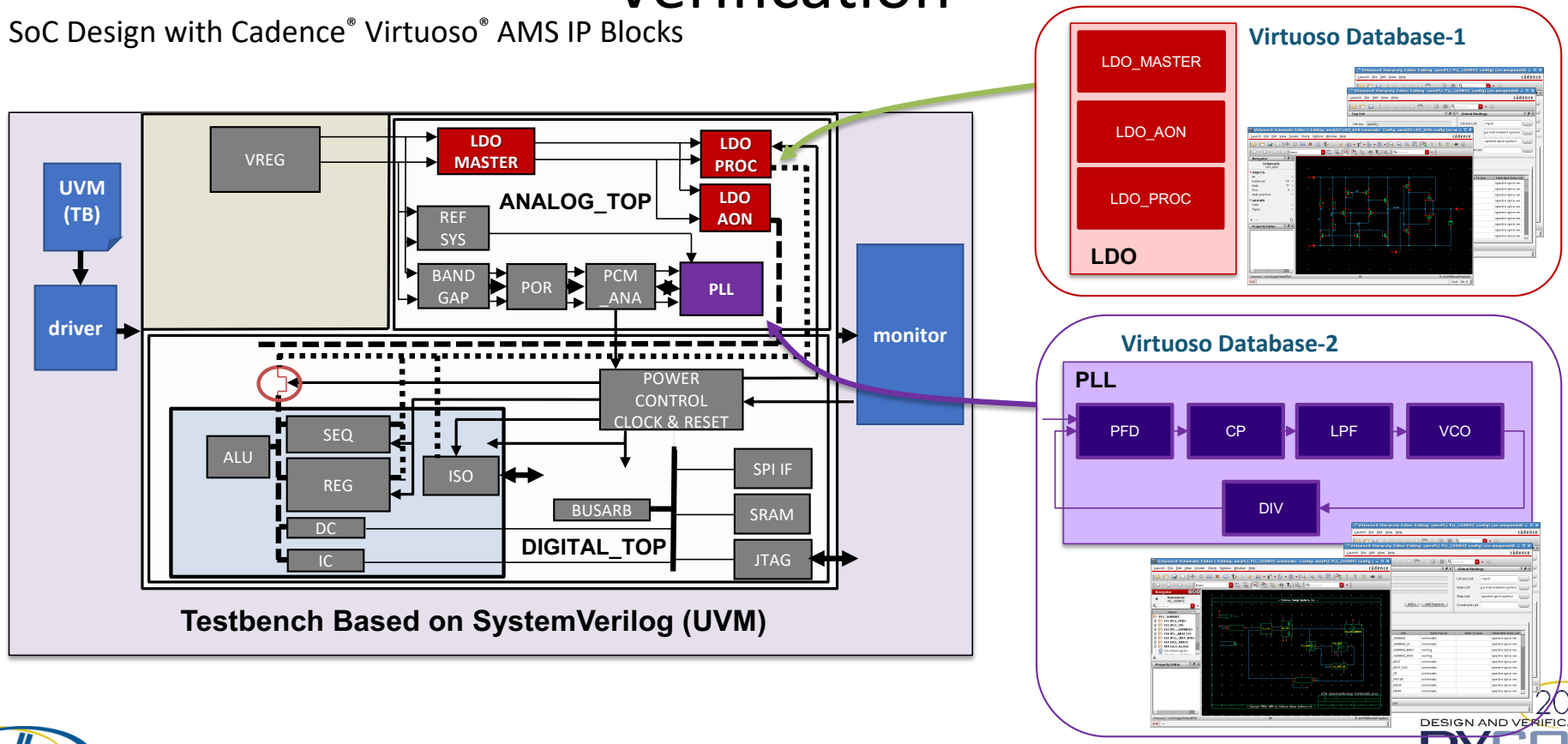
# Agenda

- Introduction
  - SoC Example architecture overview
  - Selection of flow
  - Analog core overview (Andre)
  - Digital core overview (Kawe)
- Digital Core Power Intent (Kawe)
  - IEEE 1801 introduction
  - Creation
  - Power-up verification (demonstration)
  - Power shut-off  verification (demonstration)
- SoC Power Supply  Creation (Andre)
  - Introduction
  - Step by step creation with real time demonstration
- Modeling Dynamic  Power Loading (Andre)
  - Modeling loading effect in RNM
  - EEnet intro
  - Demonstration
- Virtuoso IP Bloc Export and Reuse  (Andre)
  - Command line IP CLIPS introduction
  - Demonstration
- Using UVM-MS (Kawe)
  - UVM MS intro
  - UVM demonstration
- Conclusion and wrap-up

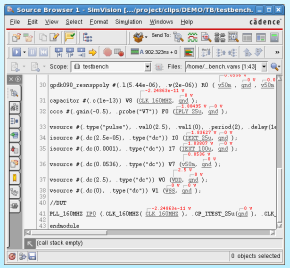


# Reuse Virtuoso AMS IP in Command-Line SoC Verification

- SoC Design with Cadence® Virtuoso® AMS IP Blocks



# Virtuoso® MS config Reuse by CLIPS



**Xcelium™ Environment**

Verification Plan

MDV | UVM/SV

Testbench

DUT (Viva/SimVision)

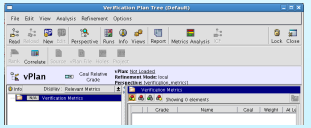
Test Plan

SV/SV-AMS

Verilog

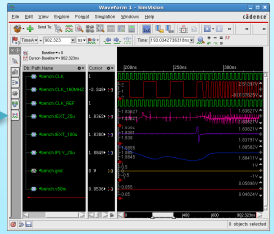
VAMS

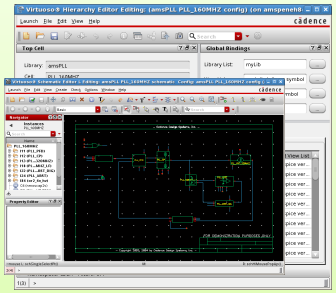
vManager™ vPlanner™



Incremental Simulation Scripts

*xrun -f xrunArgs \ -f clips.f*





VSE/HED

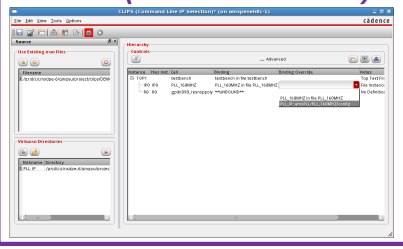
ADE | Explorer/Assembler/Verifier

Simulation (APS/XPS/AMS)

Results (Viva/SimVision)

**Virtuoso® Environment**

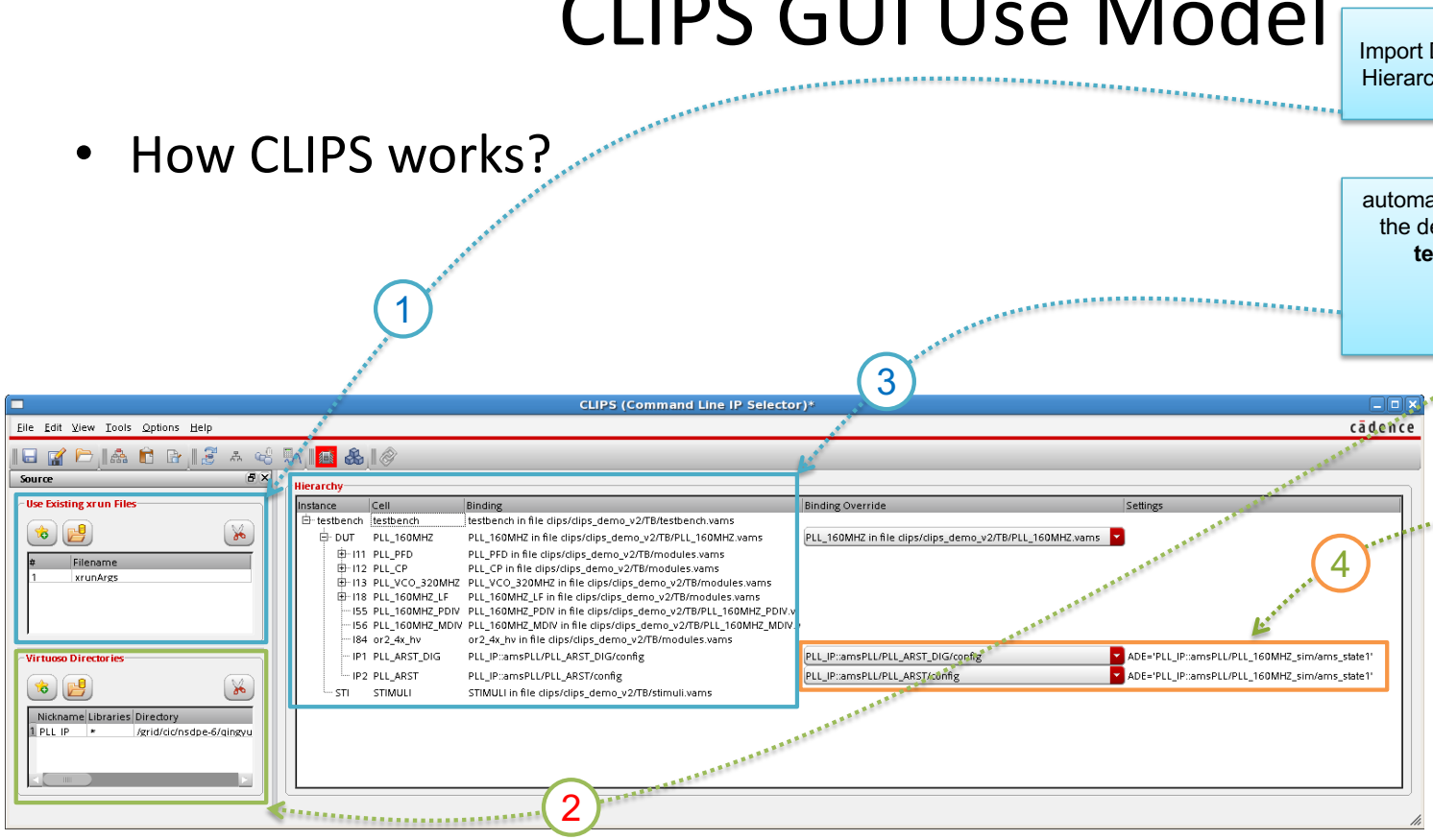
***CLIPS (GUI/Cmd-Line)***



*Simulation Files*

# CLIPS GUI Use Model

- How CLIPS works?




Import Digital Top Level Hierarchy by ***xrunArgs***

automatically elaborate the design structure
**testbench---**
**|---DUT**
**|---stimuli**
**|---...**

Import analog and mixed signal IP by ***cds.lib***

Identify the mixed signal representation in Virtuoso database

Configure binding in design hierarchy

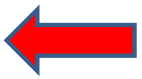
Netlisting to generate add-on clips.f file

# Virtuoso® IP Reuse for MS Verification

## Demonstration #4

# Agenda

- Introduction
  - SoC Example architecture overview
  - Selection of flow
  - Analog core overview (Andre)
  - Digital core overview (Kawe)
- Digital Core Power Intent (Kawe)
  - LP1801 introduction
  - Creation
  - Verification (demonstration)
- SoC Power Supply  Creation (Andre)
  - Introduction
  - Step by step creation with real time demonstration
- Modeling Dynamic  Power Loading (Andre)
  - Modeling loading effect in RNM
  - EEnet intro
  - Demonstration
- Virtuoso IP Bloc Export and Reuse  (Andre)
  - Command line IP CLIPS introduction
  - Demonstration
- Using UVM-MS (Kawe)
  - UVM MS intro
  - UVM demonstration
- Conclusion and wrap-up



# SV RNM: Coverage/Randomization

- Coverage/Randomization of reals
- Cadence provides full coverage/randomization support
  - Full compliment of real variable usage in randomization

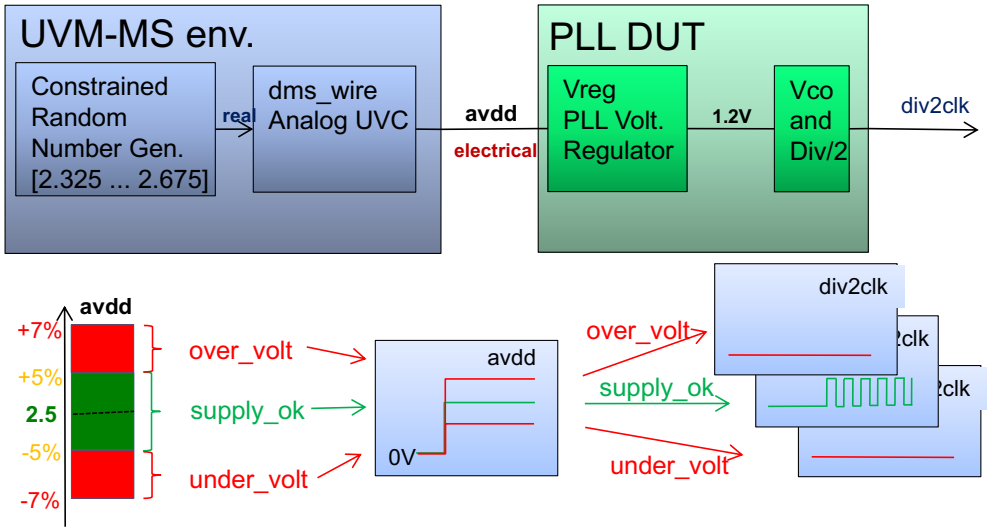
```
// Vector bins with precision
class my_tb_cls;
  rand real voltage;
  constraint my_constr {voltage dist
    { [1.0 :1.25] := 1,
      [1.25:1.5 ] := 10,
      [1.5 :2.0 ] := 1 };
  }
  covergroup cg {
    my_voltage : coverpoint voltage {
      type_option.real_interval = 0.1;
      bins b1[] = {[1.0:2.0]};
    }
  endgroup : cg
endclass
```

Randomization of the voltage

Coverage of what voltage values were generated

# PLL Mixed-Signal

## ”avdd” Supply Range Checking



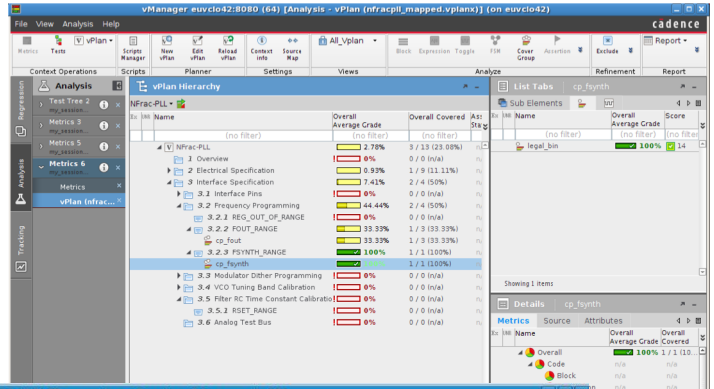
# MS Regression Control & Analysis

## Functional Coverage Results

**Covergroup definitions:**

```
covergroup bias_cg;
  bias_cp : coverpoint bias {
    bins over_volt = {[2.625:10]};
    bins supply_ok =
{[2.375:2.625]};
    bins under_volt ={[0: 2.375]}; }
endgroup // bias_cg
```

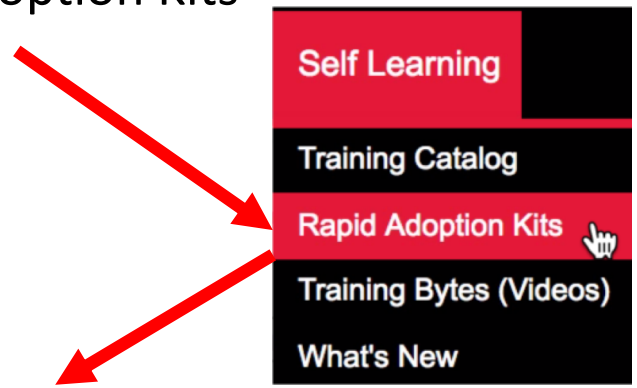
```
covergroup cg_fsynth;
  cp_fsynth: coverpoint fsynth{
       illegal_bins a =
         {[14'h2201:14'h3fff]};
       option.auto_bin_max = 25; }
endgroup : cg_fsynth
```



# Using UVM-MS
# Demonstration #5

# Gaining more knowledge …

- Access Cadence Online Support for documents and support.
  - http://support.cadence.com
- Attend Cadence Technical Training
  - https://www.cadence.com/content/cadence-www/global/en_US/home/training.html
- Download Cadence Rapid Adoption Kits
  - http://support.cadence.com




**Verification of the Power Intent of a Mixed Signal SoC (RAK)**

# Conclusion and wrap-up

- For SoC verification:
  - Avoid mixing all languages in “lasagna recipes”
  - Models only the mandatory features
- Cadence provides an comprehensive mixed-signal flow with advanced capability for creating Power Supply Network using IEEE-1801
- SV EEnet enable key capability to accurately model impedance-based interactions using Xcelium™
- Virtuoso® IP reuse increase efficiency by using CLIPS
- Cadence mixed-signal flow is an excellent solution for Low Power mixed-signal SoC design verification

# Questions

Thanks!