2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION JAPAN

Low Power Verification Using Formal Technology

Synopsys Inc.

SYNOPSYS®

accellera SYSTEMS INITIATIVE

Agenda

- Formal Verification of Low Power Designs
- Low Power Connectivity Checking
- Low Power Property Verification
- How to Overcome Formal Verification Challenges
- Summary

Formal Verification of Low Power Designs

Low Power Verification Methodology Today

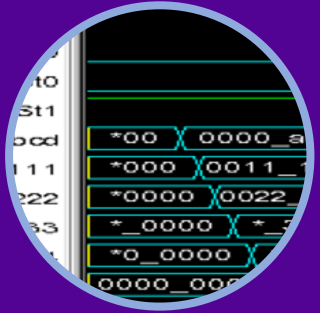
- Find Power Bugs Pre-Silicon

STATIC

- Structural checks
- Functional checks
- Architectural checks
- PG checks

SIMULATION

- Power sequence
- Low Power Coverage
- Low Power Assertions
- PST verification



EMULATION

- Complex long running power sequences
- SW PMU verification

PROTOTYPING

- Real world scenarios with real world interfaces
- SW PMU verification

Formal Low Power in the Verification Flow

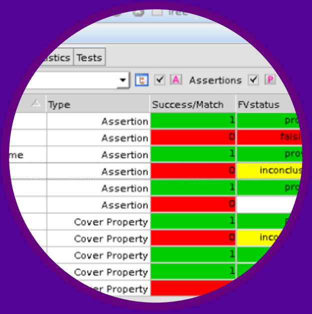
- Find Power Bugs Pre-Silicon

Static

- Structural checks
- Functional checks
- Architectural checks
- PG checks

Formal

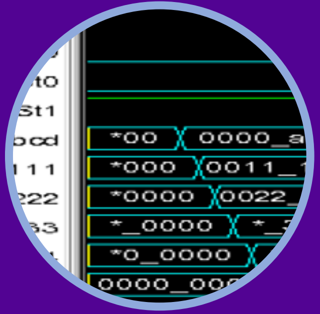
- LP Connectivity checks
- Formal LP property checks



Type	Success/Match	FVstatus
Assertion	1	pr
Assertion	0	false
Assertion	1	pro
Assertion	0	inconclus
Assertion	1	pro
Assertion	0	
Cover Property	1	
Cover Property	0	inco
Cover Property	1	
Cover Property	1	
er Property		

Simulation

- Power sequence
- Low Power Coverage
- Low Power Assertions
- PST verification



Emulation

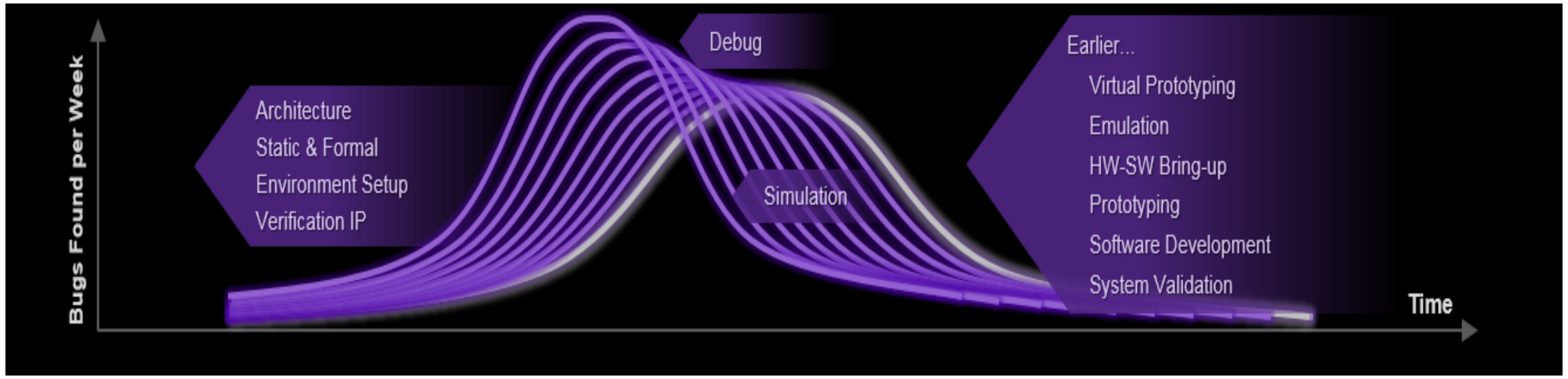
- Complex long running power sequences
- SW PMU verification

Prototyping

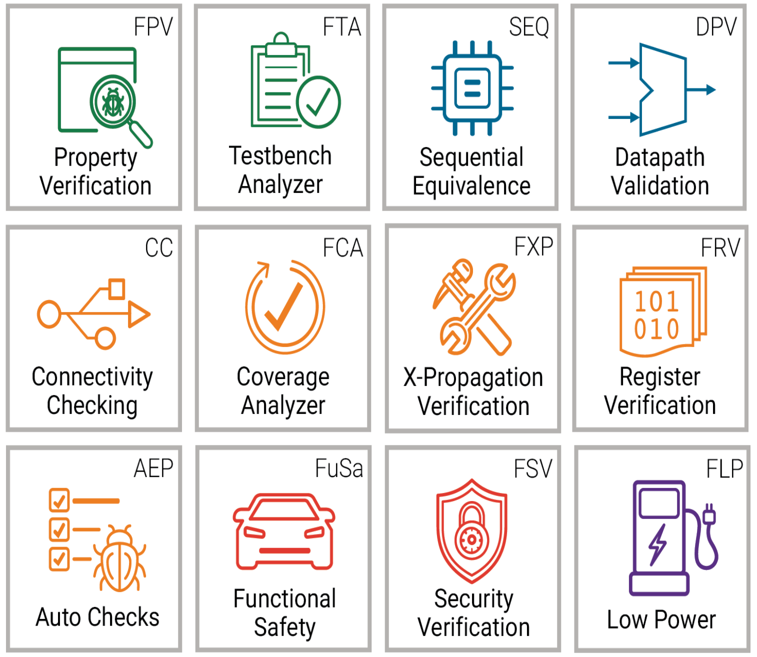
- Real world scenarios with real world interfaces
- SW PMU verification

Why Formal for Low Power

- Formal is already widely used in verification flows



Find & fix bugs as early as possible
Exhaustive verification
Find hard bugs & corner-case bugs
No testbench required
Early-stage bug-hunting



- Low power verification is an extension to functional verification

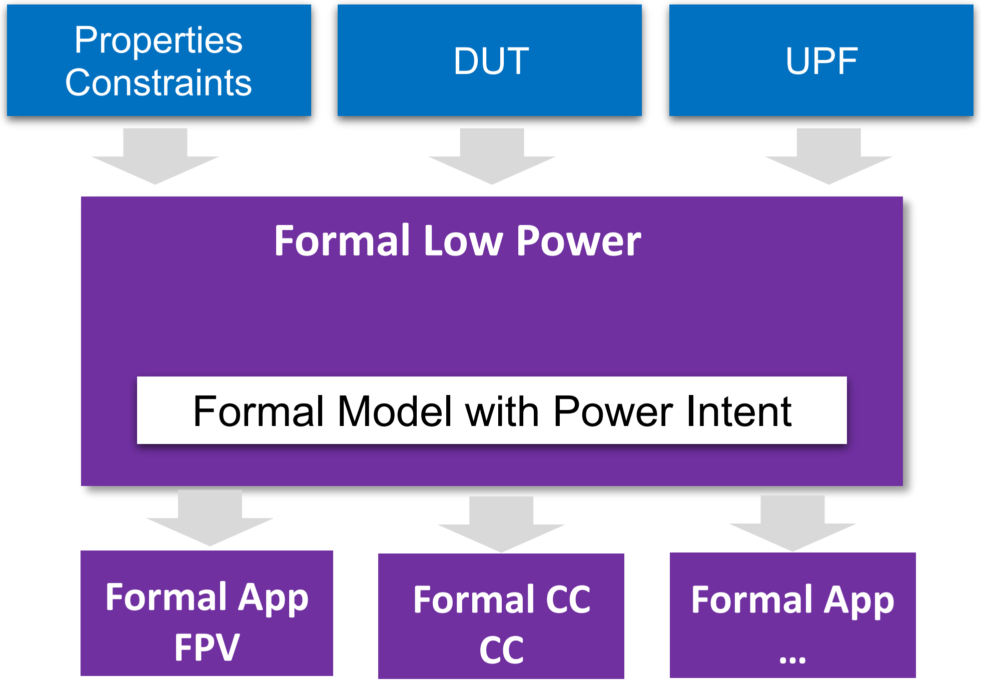
Formal Low Power Verification

FLP BENEFITS

- Start Power Aware Verification at the block level
- Complete verification of power controller
- Power aware connectivity checking with the CC App
- Power aware bug hunting with the FPV App

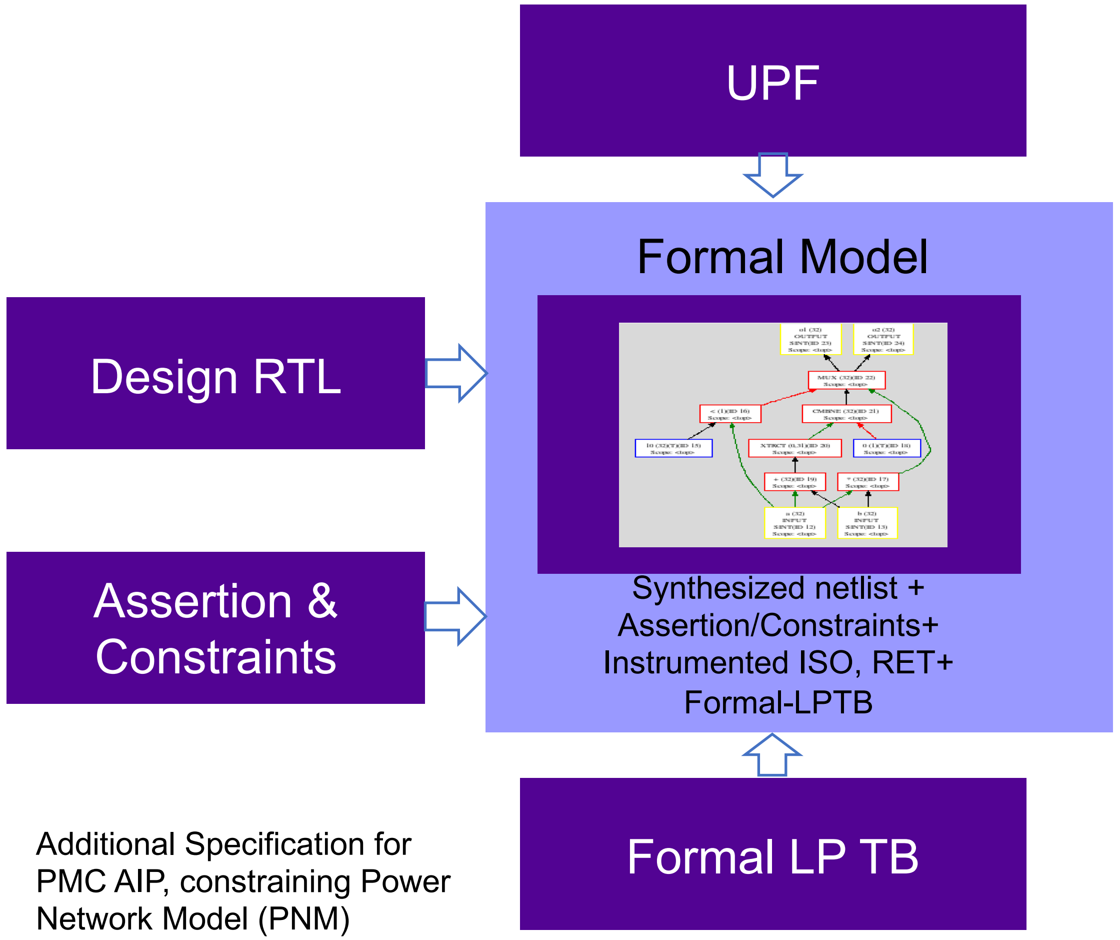
FLP FEATURES

- LP UPF compilation frontend ensures UPF interpretation consistency
- Formal low power query & assertion generation
- Check effect of Power-on-Reset sequence
- Check effect of isolation on outputs of DUT



* FLP: Formal Low Power
* CC: Connectivity Check
* FPV: Formal Property Verification

Formal Analysis of Power Aware Model



UPF
Formal Model
o1 (32) OUTPUT SINT(ID 23) Scope: <top>
o2 (32) OUTPUT SINT(ID 24) Scope: <top>
MUX (32)(ID 22) Scope: <top>
< (1)(ID 16) Scope: <top>
CMBNE (32)(ID 21) Scope: <top>
I0 (32)(T)(ID 15) Scope: <top>
XTRCT (0,31)(ID 20) Scope: <top>
0 (1)(T)(ID 18) Scope: <top>
+ (32)(ID 19) Scope: <top>
* (32)(ID 17) Scope: <top>
a (32) INPUT SINT(ID 12) Scope: <top>
b (32) INPUT SINT(ID 13) Scope: <top>
Design RTL
Assertion & Constraints
Synthesized netlist + Assertion/Constraints+ Instrumented ISO, RET+ Formal-LPTB
Formal LP TB

Additional Specification for PMC AIP, constraining Power Network Model (PNM)

1. Power Aware Connectivity Checking (CC)
   - PG Pin : Power Network connected correctly
   - Functional: Is RTL connection bug free with UPF

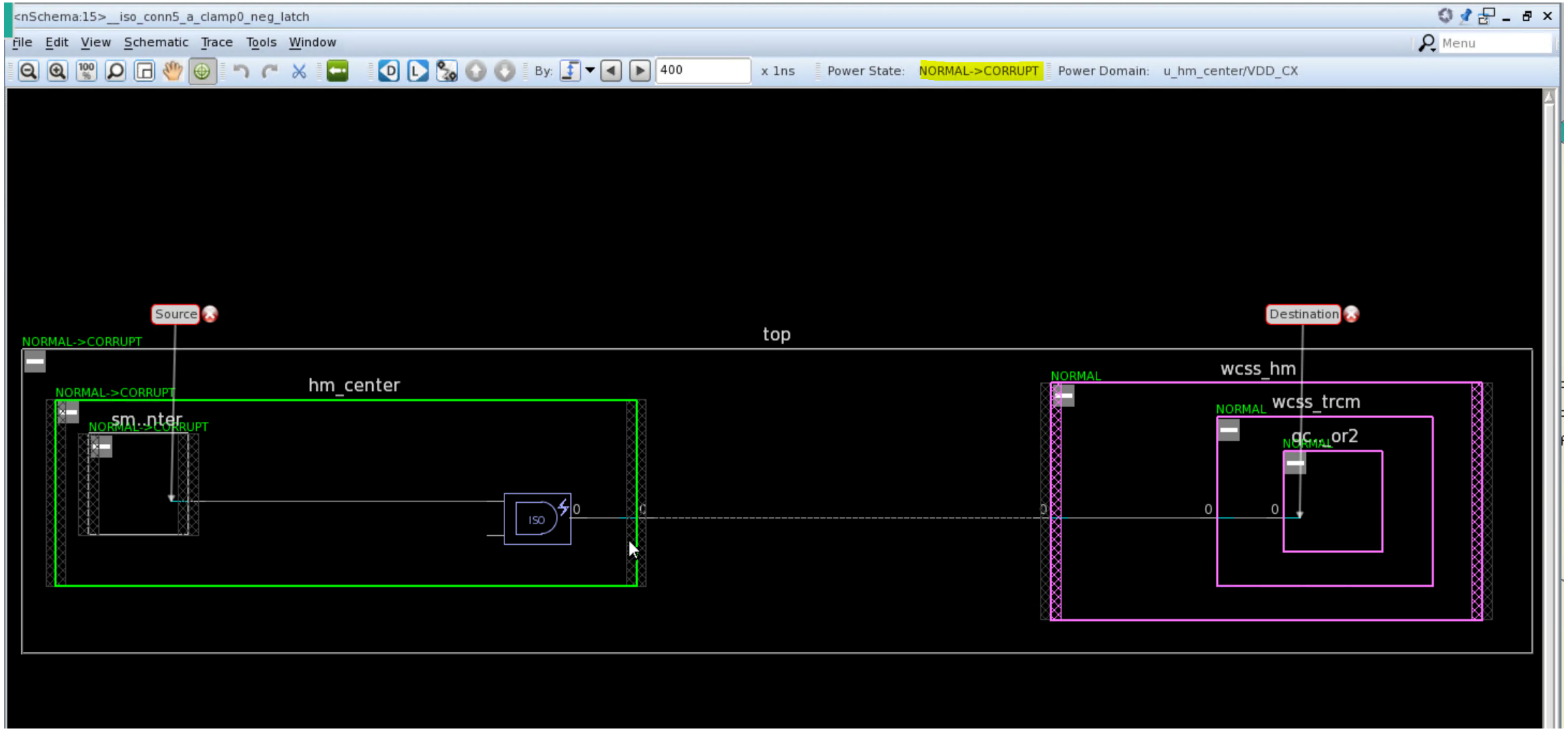
2. Formal LP Property Checks(FPV)
   - Checking effect of PoR (Power On Reset) sequence
   - Checking effect of isolation on output of DUT
   - Functional Verification of Power Management Controller (PMC)
   - Formal LP Query & LP Assertion Generation (bind_checker)

Formal Low Power Applications

- Power aware connectivity checking at SoC level
- Corner-case low-power verification with formal technology
- Verifying power-controller is working correctly with the UPF provided
- Ensuring power-on-reset getting the design back into a known state
- Exhaustive formal verification of retention and isolation control/data paths at block/IP level

Low Power Connectivity Checking

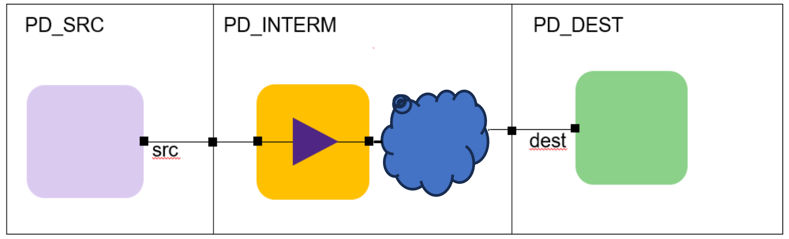
Low Power Connectivity View



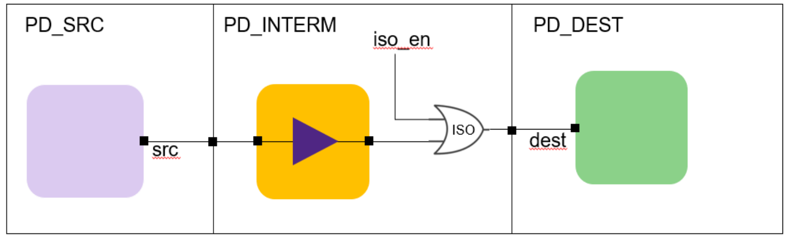
<nSchema:15>__iso_conn5_a_clamp0_neg_latch
File Edit View Schematic Trace Tools Window
Menu
By:
400
x 1ns
Power State: NORMAL->CORRUPT
Power Domain: u_hm_center/VDD_CX
Source
Destination
top
NORMAL->CORRUPT
hm_center
NORMAL->CORRUPT
sm..nter
NORMAL->CORRUPT
ISO
wcss_hm
NORMAL
wcss_trcm
NORMAL
qc...or2
NORMAL

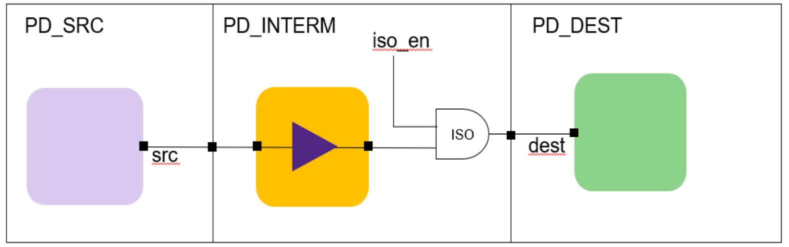
Low Power Connectivity Scenarios

- **How to check connection passing through always on domains (LPA BASIC)?**
  - Expect no isolation cell in the path
  - Both power domains, PD_SRC PD_DEST are NORMAL
  - `En && SD && DD |-> (dest == src)`



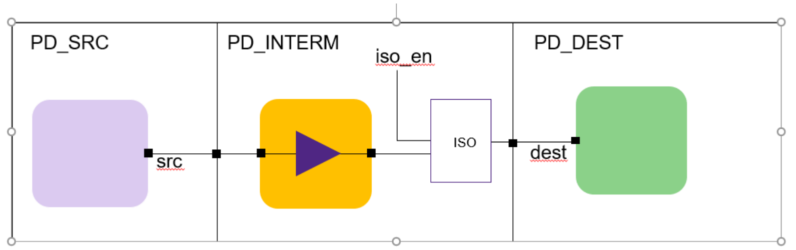
- **How to check connection passing intermediate powered off domain with type of iso cell (LPA_CLAMP1/CLAMP0)?**
  - CLAMP1 is OR-type ISO cell & CLAMP0 is AND-type ISO cell
  - `En && SD && DD && (iso_en != ISO_SENSE) |-> (src == dest)` … (**`connectivity component`**)
  - `En && SD && DD && (iso_en == ISO_SENSE) |-> (dest == 'b1)` … (**`clamping component`**)



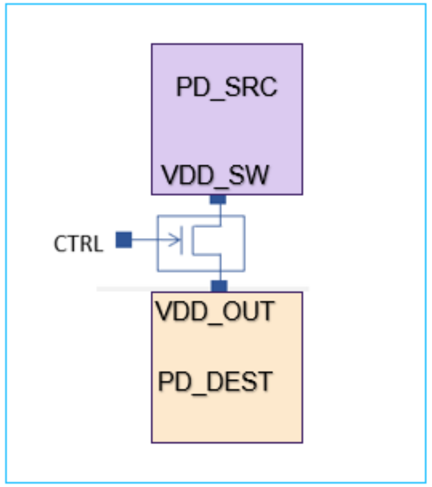


Low Power Connectivity Scenarios

- **LPA_LATCH**
  - Same as LPA_CLAMP1/0 but having LATCH-type iso cell.
  - Consequent of the clamping component becomes (not $fell(dest) and not $rose(dest)).



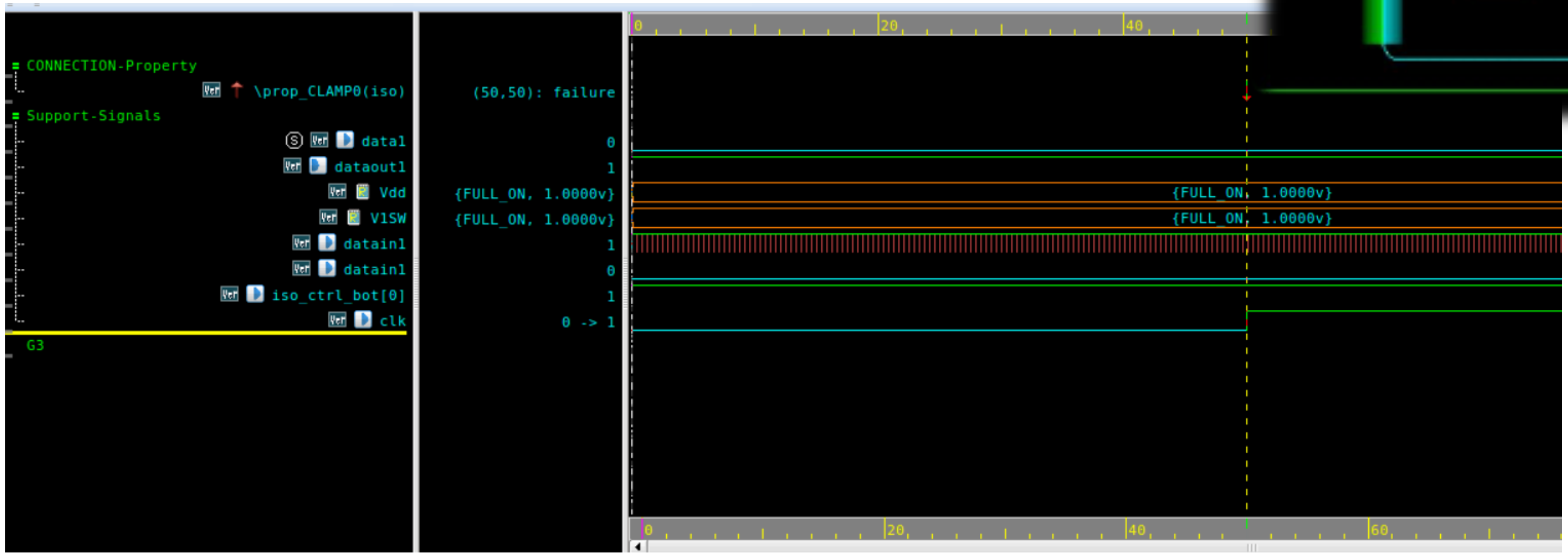
- **LPA_SUPPLY**
  - Power/ground (PG) pin connectivity checking.
  - Source & Destination should be power supply objects from the UPF.



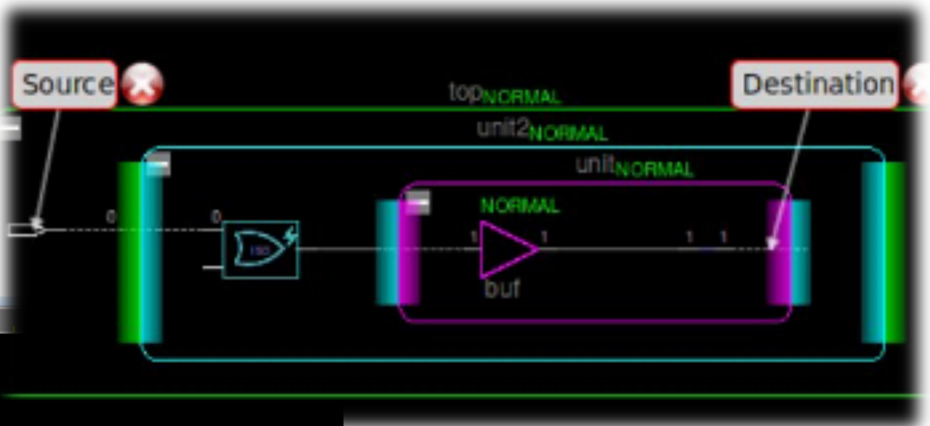
- **LPA_CLAMP1_EN/CLAMP0_EN/LATCH_EN**
  - Connectivity of enable source to enable pin of instrumented isolation cell
  - `En && SD && DD && (src == ISO_SENSE) |-> (dest == 'b1) (clamped)`

Example of FLP CC Bug

- Broken connection
  - Source: data1
  - Destination: u2.inst_unit.dataout1
- **Cause: LPA_CLAMP0**



```
CONNECTION-Property
  \prop_CLAMP0(iso)      (50,50): failure
Support-Signals
  data1                  0
  dataout1               1
  Vdd                    {FULL_ON, 1.0000v}
  V1SW                   {FULL_ON, 1.0000v}
  datain1                1
  datain1                0
  iso_ctrl_bot[0]        1
  clk                    0 -> 1
G3
```



* FLP: Formal Low Power
* CC: Connectivity Check

Case Study: FLP Connectivity Checking

- Catching Bugs Earlier Shortens Project Cycle

Design	Low Power Formal Checks	# of Bugs Found	Benefits
Design 1	Isolation mismatch between spec and UPF	100+	Verified in 1 day.
Design 2	PG pin connectivity checks	2+	Verified in ½ hour
Design 3	Isolation signal propagation checks	1+	Verified in 1 day

* FLP: Formal Low Power

Low Power Property Verification

Power Aware Formal Property Verification

- What is Power Aware FPV
  - It is checking LP behavior of the PA-RTL using PA assertions
  - Formal checking of power aware reset sequence
  - Checking for X propagating due to incorrect low power behavior

- What is the Purpose of Power Aware FPV (PA FPV)
  - Power intent UPF design can be tested in any existing FPV testbench
  - Properties failure point to LP issues in the design
  - Provide GUI based debug platform for LP issues
  - Shift left PowerOnReset functional verification

- Leverage existing NLP:bind_checker infrastructure for FLP FPV

* FLP: Formal Low Power * PA: Power Aware * FPV: Formal Property Verification

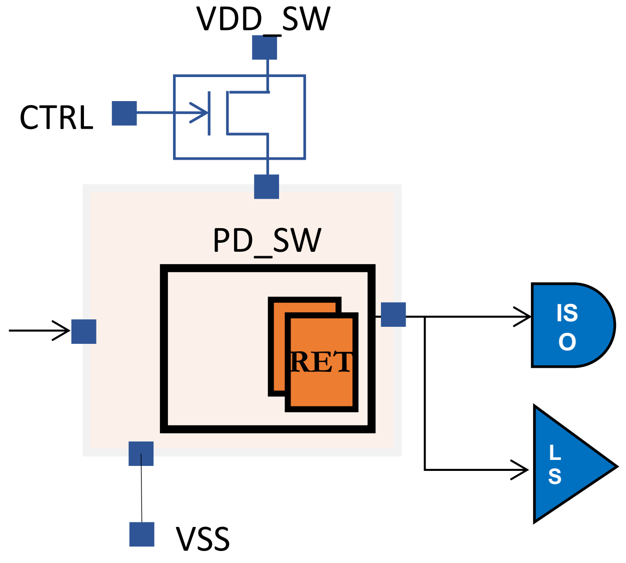
Formal Low Power Property Verification

FLP BENEFITS

- Start Power Aware Verification at the block level
- Complete verification of power controller
- Power aware bug hunting with the FPV App

FLP FEATURES

- Formal low power query & assertion generation
- Check effect of Power-on-Reset sequence
- Check effect of isolation on outputs of DUT



- Is my power controller working correctly with the UPF provided?
- Are my isolation clamp values correct?
- Are my retention signals ordered correctly?
- Are there signals becoming unknown due to upstream power domain powering off?
- Does the power on reset get the design back into a known state?

* FLP: Formal Low Power * FPV: Formal Property Verification

Low Power Property Examples

- **Checking effect of isolation on output signal of DUT**

Iso enable

Isolation output should clamp to value 1

```
property p_isolation_check1;
disable iff(rst )
  @(posedge clk)
    ##1 (SRC_PD_en ==0) &&(clamp_enable==1) |=> (ISO_element_OUT ==1)   ;
Endproperty

isolation_check1 : assert property (p_isolation_check1);
```

- **Checking effect of reset sequence on power up**

Check signal value is equal to reset value

Power collapse exit ,clock disable and reset state enable

```
property check_reset_values_after_power_up ;
     disable iff(rst|| ((!SRC_PD_off)) ) @(posedge clk)
          ((pwr_collapse_en==0 && CLK_gate_dis_ack==1) && (t_rst) )|-> (qacceptn) === reset_valur));
endproperty

qreqn_check_reset_values_after_power_up : assert property   (check_reset_values_after_power_up ));
```

- **Check if sequential logic is uninitialized on wakeup, propagating through:**

Should not get unknown values

Power domain control signals ==0

```
property p_tx_isunknown_check_0;
disable iff(rst )
  @(posedge clk)
    ##1 (SRC_PD_en ==0) |=> not ($isunknown(qnm_rot_Rsp_U_Tx[0] ) ) ;
endproperty

tx_isunk_check_0 : assert property (p_tx_isunknown_check_0);
```

Case Study: FLP Formal Property Verification

- Catching Bugs Earlier Shortens Project Cycle

Design	Low Power Formal Checks	# of Bugs Found	Benefits
Design 1	Isolation is not enabled in power shutdown	10	Verified in 1 Day

* FLP: Formal Low Power

How to Overcome Formal Verification Challenges

Challenges in Formal Low Power App

- SoC size to verify low power in FPV
  - UPF created in chip level verification
  - Power management controllers at the top level
- Lack of existing FPV testbench
- Lack of knowledge in BOTH formal and low power
- Lack of connectivity specification

* FPV: Formal Property Verification

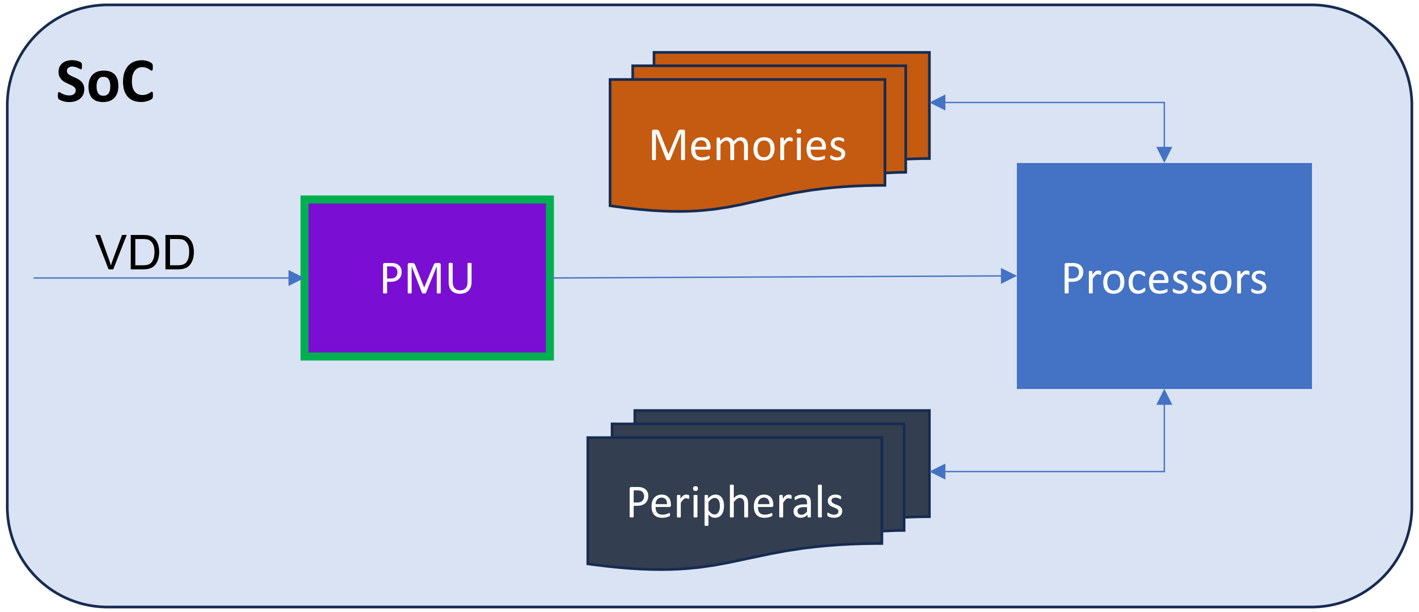
Techniques to Overcome SoC CC Challenge

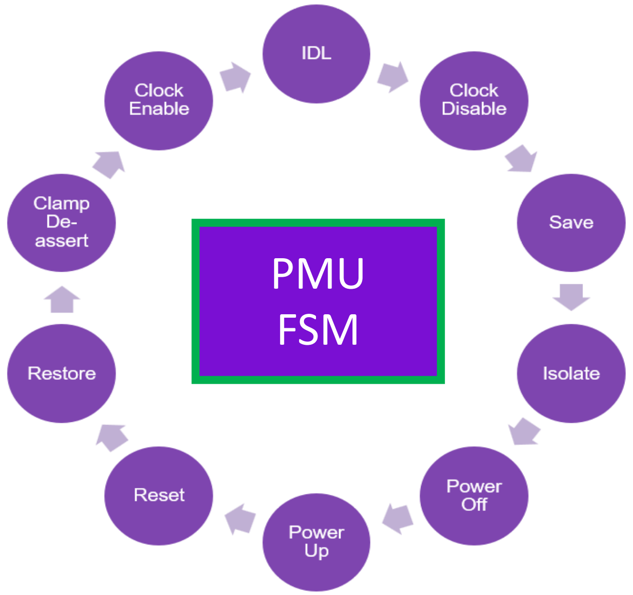
- Identify elements to blackbox before design compilation
  - Memory blocks
  - FIFOs
  - Use tools auto-blackbox feature
- Do not combine retention instrumentation in FLP Connectivity Checking
  - Run retention checking separately

* CC: Connectivity Check

Techniques to Overcome SoC FPV Challenge

- Power management controllers
  - Verify at the block level
  - Create at IP level based on their UVM power control logic
  - Review simulation waveform and create FSM for PMU.





* FPV: Formal Property Verification

Reduce Complexity for Better Convergence

- Identify elements to blackbox before design compilation
  - Blocks without isolation cells
  - Blocks in always on domain
  - Memory blocks
  - FIFOs
- Abstract complex logic
  - Counters
- Run retention instrumentation separately from FPV

* FPV: Formal Property Verification

Overcome Other FLP Challenges

- FLP connectivity checking
  - EDA vendors like Synopsys can provide example connections and test cases for different scenarios
  - Work with design architect for critical paths and scenarios to test design and create connectivity spec
- FLP FPV checking
  - Synopsys can provide sample FPV Assertion test bench
    - Additional scenarios by user adding more SVA properties
  - Recommend at least basic training for both Formal Apps and Low power before applying

* FLP: Formal Low Power
* FPV: Formal Property Verification

Summary

Summary

- Low power verification using formal enables shift left
- Formal verification Apps for connectivity check is easy to use
- The failure trace for debug are short
- Bugs can be found very quickly
- Apply reducing complexity and divide and conquer techniques to over come the challenges in the formal low power FPV verification

* CC: Connectivity Check

Thank You