2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Efficient hierarchical low power verification of custom designs using static and dynamic techniques

Himanshu Bhatt (himanb@synopsys.com)
Synopsys, Inc. (US)
Archanna Srinivasan (archanna.srinivasan@intel.com)
Intel PSG (US)
Chong Lee Kuay (chong.lee.kuay@intel.com)
Intel PSG (Penang)

Abstract-As custom SoCs (ASIC + FPGA) are gathering momentum because of the ease of custom programming, low power verification is becoming increasingly imperative. The biggest challenges faced are that the industry leading tools for both dynamic and static verification are not tailor-made for the custom based designs. The custom designs start from the schematic which is then written out into a netlist using certain tools. Hierarchical UPF verification for such SoCs is challenging. Also, there is no defined methodology for low power verification of custom blocks. The paper describes a methodology for an efficient hierarchical low power verification ensuring that subtle bugs do not escape into silicon.
Keywords: SoC (System on Chip), UPF (Unified Power Format), BMOD (Behavioral Model), ASIC (Application Specific Integrated Circuit), FPGA (Field-programmable Gate Array), TAT (Turn Around Time)

I. INTRODUCTION

With increasing SoC complexity, the need for a robust low power verification is important for signing off the overall verification. For custom SoCs, there are various challenges since neither there is a well-defined low power methodology that exists, nor the industry leading low power verification tools are tailor-made for such custom designs. This opens an opportunity to define a robust methodology to effectively verify the low power intent for such designs. This paper describes the combination of static and dynamic low power verification for verifying such custom SoCs.

II. LOW POWER VERIFICATION CHALLENGES

There are many challenges with low power verification

- Huge verification space
  - Large number of power states (based on the power modes)
  - Large number of transitions
    - Software applications
    - Firmware
    - Digital Hardware
- System level verification
  - Reuse in larger system (most SOCs are derivative so must ensure reuse of methodology)
  - Often requires HW/SW co-simulations/emulation/prototyping (PMU being part of the S/W)
- LP specification extensive
  - New versions (onus on the EDA vendors to ensure the consistent UPF interpretation of new constructs)

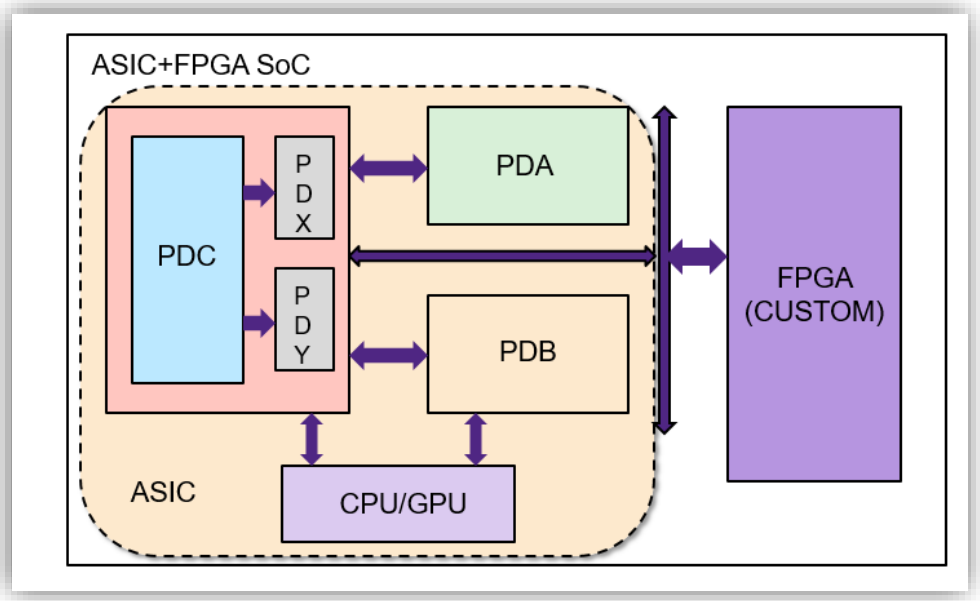

Figure 1: ASIC+FPGA SoC

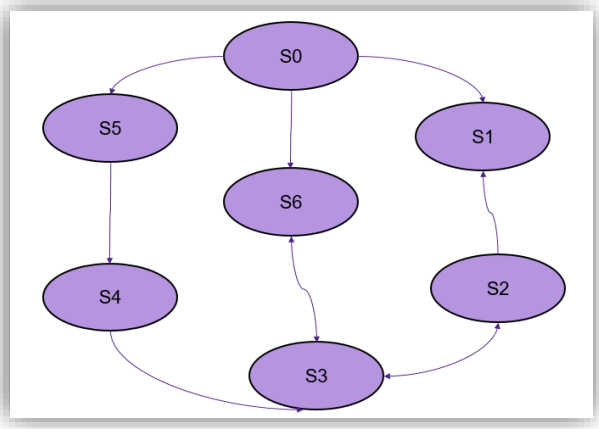

Figure 2: Multiple Power States (Power modes)

Figure1 above shows an ASIC+FPGA SoC. There are multiple power domains which necessitate the usage of special cells such as isolation, retention, level shifters, AON cells and power switches. The large number of PSTs poses a challenge to verify all the legal transitions comprehensively. With the low power structures needed, the number of scenarios needed to verify rises exponentially when compared with a traditional non low power functional verification.

III. PROPOSED METHODOLOGY

Effective verification with UPF is very important since there is no “one solution that fits all”. There are certain issues which are better to be caught upfront even before going to the simulation stage to reduce the overall verification TAT. A static low power verification methodology was deployed for this case. Power sequencing, power control, retention modeling and coverage was verified using a power aware simulator. The methodology will

discuss the different aspects of low power verification for the ASIC and FPGA designs using the combination of static checker and power-aware simulator.

LOW POWER VERIFICATION FLOW

The figure below shows the ASIC low power verification flow which is widely used across the industry.

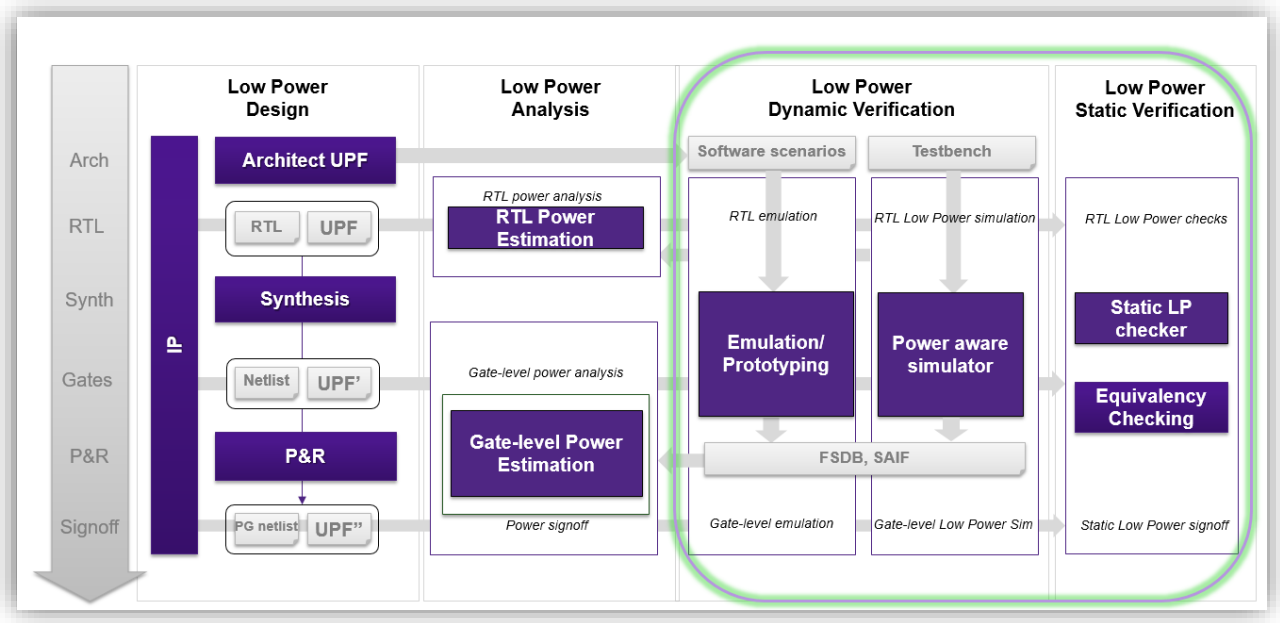

Figure 3: Low Power verification flow

Below is a summary of which category of bugs are better caught by each of these tools (static and dynamic)

- What is checked better using static checker?
  - Are my isolation and level-shifter policies complete?
  - Am I missing protection?
  - Are my power nets connected correctly?
  - Does my power implementation match my power intent (UPF)?
- What can functional simulation with UPF catch?
  - Is my power-controller working correctly with the UPF provided?
  - Are my isolation clamp values functionally correct?
  - Are my retention signals ordered correctly?
  - Are signals unexpectedly becoming unknown due to some upstream power domain turning off?

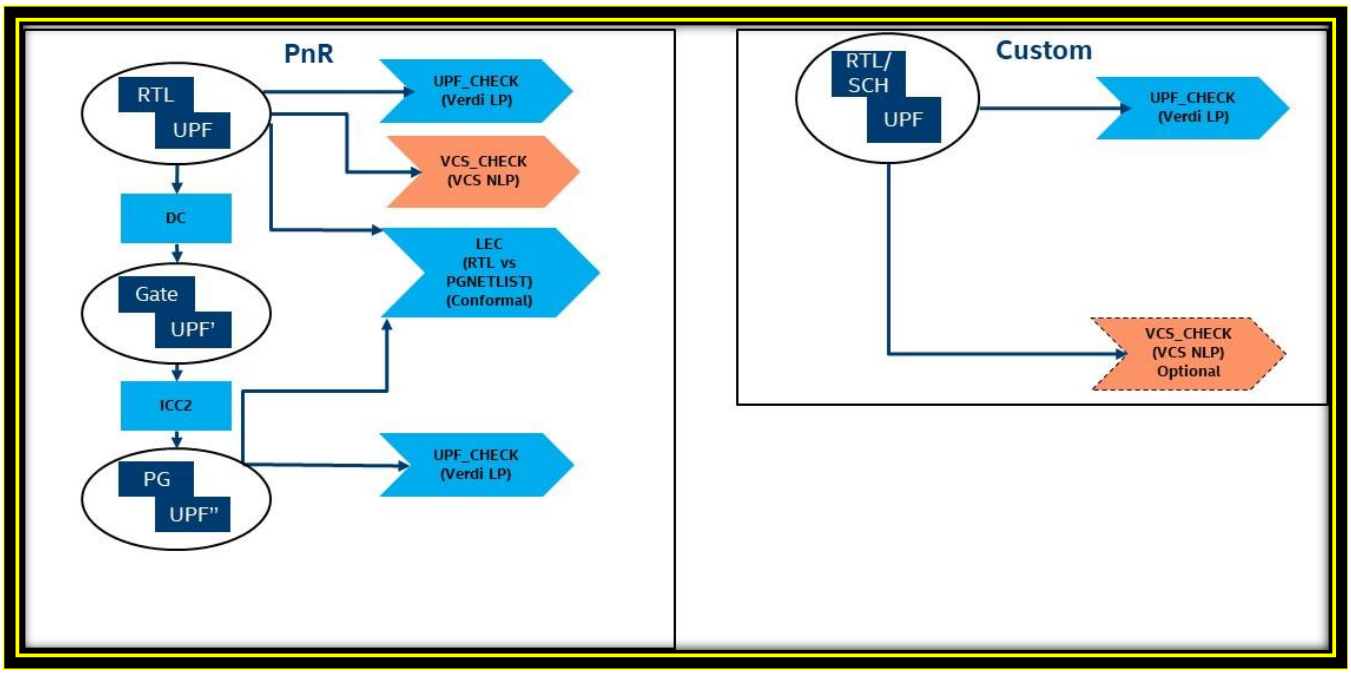

Figure 4: Low Power verification flow for PnR (ASIC) and Custom designs

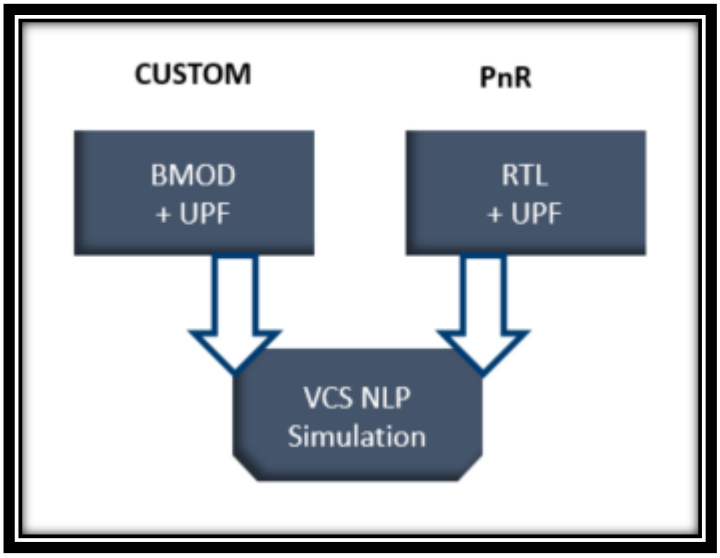

Figure 5: Power aware simulation for PnR (ASIC) and Custom designs

For Custom Covering ASIC, listed are two approaches in UPF power intent verification. As the top-level Custom block UPF is only used in verification, it has no constraint that we encountered in ASIC implementation flow. The power intent verification can either be done using timing library or PG netlist delivered by the encapsulated ASIC

block. The UPF of the encapsulated ASIC design is ready and verified using power intent checking tools. The same UPF has gone through ASIC design implementation flow and it can generate the PG netlist and timing library after the design implementation. The Custom-ASIC hybrid block UPF flow using ASIC timing library is similar to the approach used in ASIC-Custom hybrid block. In this case, the top level Custom UPF does not load the encapsulated ASIC block UPF. The Custom top level UPF only include its power intent definition until the ASIC block boundary. The connectivity of the power supplies from the top-level Custom block that required by the ASIC design must be included in the top-level Custom block UPF. This is to ensure the ASIC block receives the required power connected to its PG pin. The Custom-ASIC hybrids block UPF flow using ASIC PG netlist is similar to the conventional hierarchical UPF flow. The UPF of top-level Custom design loads the ASIC block UPF. In this case, the power intent verification is performed completely on the entire Custom-ASIC hybrid block.

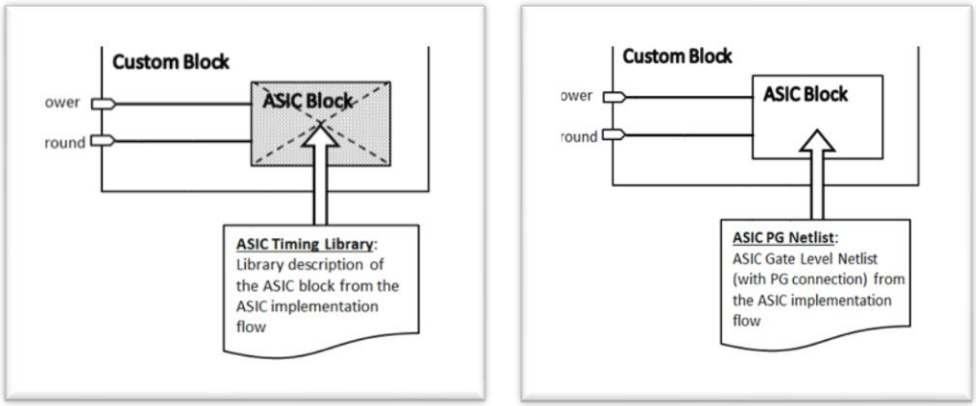

Figure 6: Custom encompassing ASIC

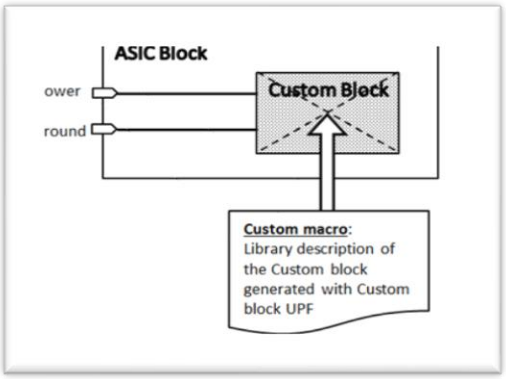

Figure 7: ASIC encompassing Custom

One most important aspect of hierarchical low power verification using UPF is the *.lib/.db creation for lower level blocks Once the dynamic and static low power verification is signed off for each custom block, a .lib/.db is created for it. For all such custom design blocks, a .lib/.db is similarly created. Figure 8 below provides the details of these .lib/.db creation.

The ASIC low power verification methodology is well established. Figure 8 shows the methodology to verify custom design blocks using both static and dynamic verification methodology. For dynamic low power simulation, a behavioral model (BMOD) is created by the schematic designer. The power aware simulator uses this BMOD along with the UPF to perform various low power dynamic checks and provides PST coverage. The schematic is converted to a Verilog gate level netlist and the static verification tool uses this netlist and the same UPF to perform static low

power checks. Once the dynamic and static low power verification is signed off for each custom block, a .lib/.db is created for it. For all such custom design blocks, a .lib/.db is similarly created.

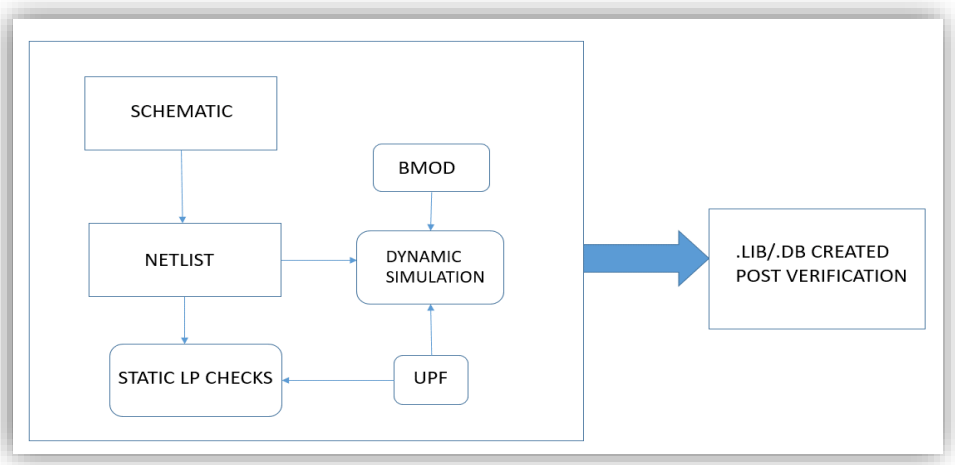

Figure 8: Custom blocks verification

Figure 9 depicts the flow for the creation of custom .libs and .dbs. In case the .lib is available which meets the designer requirement for MV cells (ISO/RET/PSW), a .db is created directly for such .libs.
In cases where the designer intent for MV cells is not available in the available .libs, manual .libs are created which are then converted to .dbs

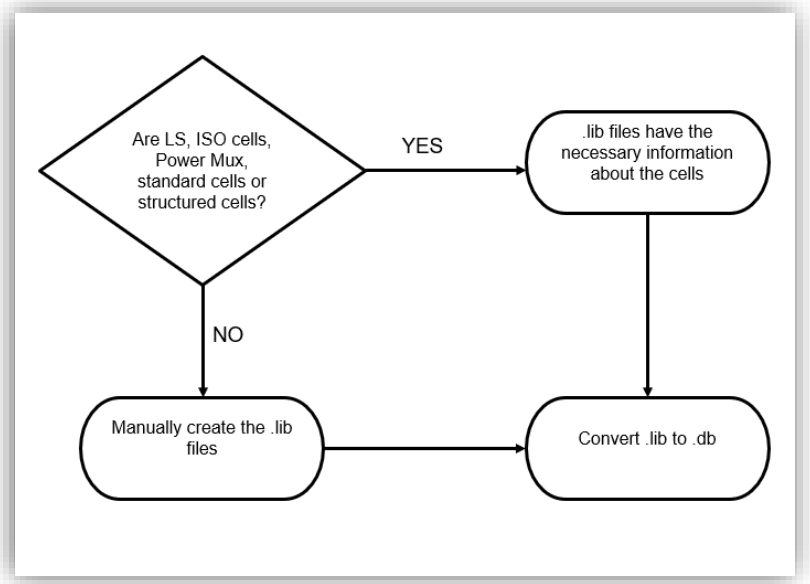

Figure 9: .lib/.db creation

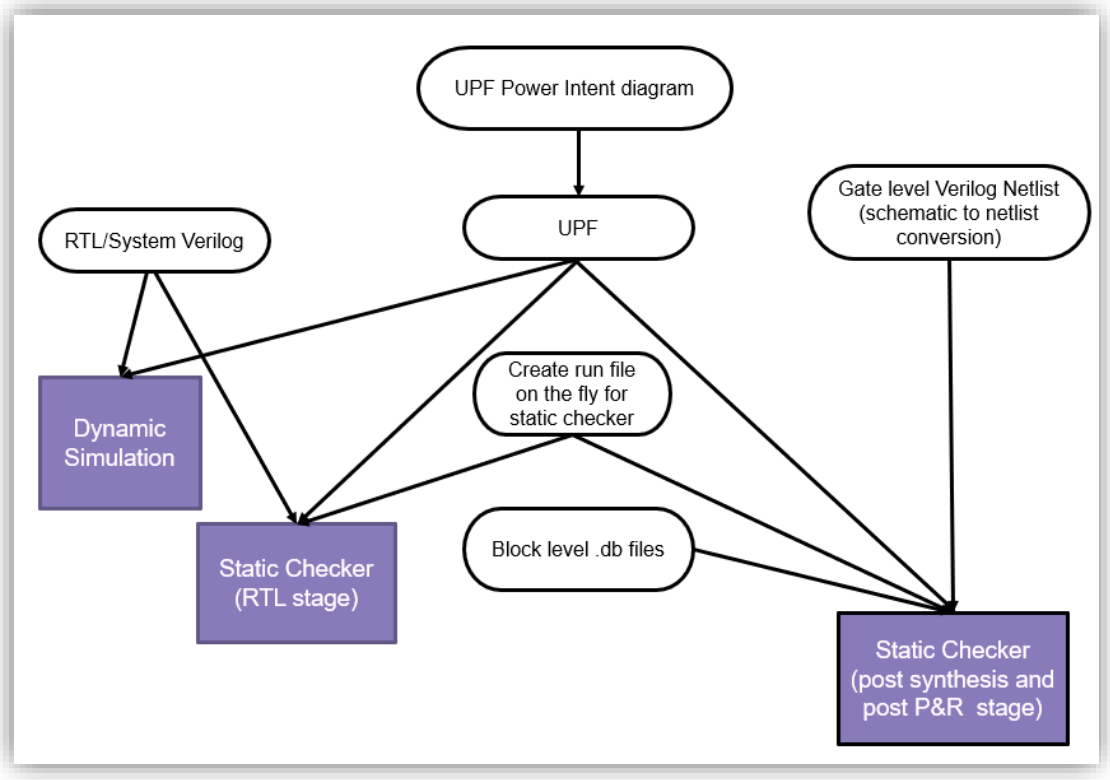

Figure 10 : Overall flow diagram for Low Power verification

At the sub-system level, a sub-system level UPF is created and all the lower level .lib/.db which are referred to as custom macros are then loaded (using the black-box methodology).

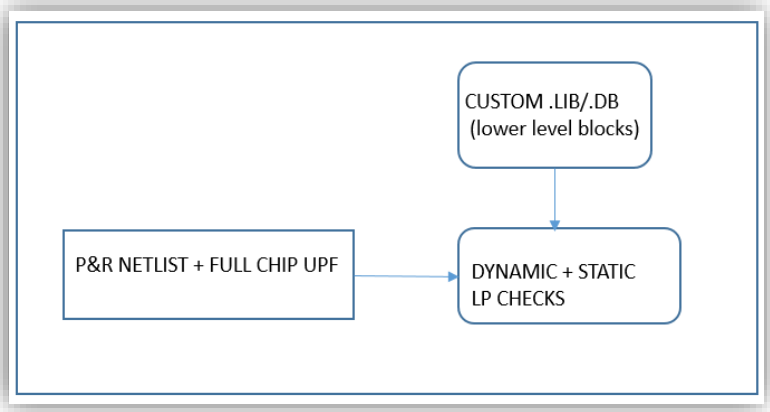

Figure 11: Full chip verification

The top level in this hierarchy is an ASIC block. This ASIC block goes through the synthesis and place and route stages and a P&R netlist is generated along with the full chip UPF. Both static and dynamic verification is then run on the full chip P&R netlist with the loading of the .lib/.db of the sub-system custom block as shown in Figure 4.

IV. Results And Conclusions

Using the hierarchical low power verification methodology described above, enabled to catch some critical issues in the designs (details cannot be divulged since it is Confidential Intel data). As mixed-signal and adaptive computing platforms are gathering momentum, low power verification is becoming increasingly challenging. The biggest challenge faced is that industry tools for both dynamic and static verification is not tailor-made for these custom designs. The custom designs start from the schematic and written out into a netlist using certain tools. Power intent creation for these netlist designs is challenging as there is no defined methodology for low power verification of custom blocks. The combined Synopsys static and dynamic low power verification methodology for mixed-signal designs ensures that real life bugs do not escape to silicon

	Power Management Bug/Issue	Synopsys Tool
*1	Structural Errors	Covered by VC-LP (Static Check) -> (*1)
1.1	**Isolation and Related bugs**	
	Missing Isolation	PST merging
	Incorrect Isolation Polarity	Architectural check without stimulus
	Incorrect Isolation Enable Polarity	UPF power consistency check between power domains
	Incorrect Isolation Gate Type	Missing/redundant LP strategy
	Redundant Isolation	
	Gated Latch	LP cell location check
	Ungated Latch	LS shifter check
	Pull-ups/pull-downs	Isolation wrong clamp value vs. ISO cell lib function
1.2	**Level shifter and related bugs**	Power/voltage crossing checks
	Level shifter out of range	
	Incorrect domain of level shifter	
1.3	**Others**	
	Power switch structure	
	PG connectivity checks	
	Hard macro internal power that have multiple domains	
*2	Control/Sequence errors	Check covered in VCS-NLP (Dynamic check) -> (*2 & *3)
2.1	**Isolation control errors**	Legal Power States Check (PST)
	Incorrect Isolation Enable Timing	LP sequencing & transition related checks
	Redundant Isolation	X-propagation
	Memory Corruption in Standby (unsafe write)	Covergroup (PST coverage)
	Save and Restore Sequences	Power-on-reset check
	Power wastage due to control error	Verify PMU control signal legal state during transition (ISO/Retention/Power Switch)
2.2	**Logic Corruption**	Isolation wrong clamp value affecting functionality
*3	Architectural Errors	Corruption affecting functionality on ON domain
	Power Gating Collapse	Automated assertions
	Memory Corruption in Standby (COA, CSOC)	LP control signal corruption check
	Cold boot	LP sequencing & transition related checks

Figure 12: Verification sign-off using low power static and dynamic tools

Acknowledgment

The authors would like to thank the entire Intel PSG Design and Methodology team for creating this methodology and taking it for adoption among all the DV teams.

References

[1] 1801-2015 - IEEE Standard for Design and Verification of Low-Power, Energy- Aware Electronic Systems
[2] https://readytalk.webcasts.com/starthere.jsp?ei=1220777&tp_key=33674ddc63&sti=cp&elq_mid=10720&elq_cid=158897
[3] https://semiengineering.com/overcoming-low-power-verification-challenges-for-mixed-signal-soc-designs/