

Combining Static and Dynamic Low Power Verification for the Power-Aware SoC Sign-off

Himanshu Bhatt
Corporate Applications Engineer
Synopsys, Inc.
himanb@synopsys.com

Prashanth M
Corporate Applications Engineer
Synopsys, Inc.
mprash@synopsys.com

The low power design size and complexity is increasing tremendously over the years. Engineers are always hard-pressed with time-to-market concerns. While high-level specifications hold a promising future, RTL is the commonly used representation of the design intent today. RTL guides the implementation process, and relies on verification process to make sure that the power intent is carried through to silicon. In a power-aware design, the power format file like IEEE 1801 (UPF) is the specification (and not the RTL by itself) for the power intent. Since all the tools in the end-to-end verification flow can read the power intent, it becomes tricky for the verification engineers to decide the type of the low power verification methodology that needs to be used for the power-aware designs.

I. INTRODUCTION

Since all the design flows in the low power verification solution read the same power format files and RTL, engineers would assume that information integrity is sustained. Therein lies the problem. Logic verification is an abstraction of the design in which the power rail connectivity and physical placement, among other details, are not important. The UPF 1801 and other standards are designed to ensure that the tools have the same user-defined intent, but that does not guarantee that they have interpreted that information correctly. While the same problem exists for RTL to some extent, the collective experience over a decade has resolved the discontinuities. In today's context, the pace of the low power design, the market, and cost of silicon failure do not allow the electronics industry a similar "baking" time for the power-aware design.

Isolation, retention, and power switches are the important functionalities of the power-aware designs which use some of the common low power techniques like power shutoff, multi-voltage and advanced techniques like DVFS, Low VDD standby, and biasing. The

strategies for isolation, retention, and level shifter are specified in the power format file. In the dynamic low power verification, simulator reads the power format file during RTL elaboration and evaluates the isolation and retention information in preparation for the simulation test. The aim is to make sure that the sequences for shut down, isolation, and retention are accurate. Another important aspect of a dynamic simulation is the usage of the techniques such as assertions for low power sequences, correct clamp value for isolation, save/restore handling with reset, power-on handling, multi-rail macro handling, and coverage of the power states.

Designers usually specify the protection devices in the RTL. There should not be a condition where the global signals (clock, reset, and test_en) or control signals (isolation and power switch enable) cross from the source domain to the sink domain, and source domain ON condition is not a superset of the sink domain ON condition. If these conditions exist in the design, they can cause functional issues because when source is OFF, signal cannot reach its destination domain. It becomes an important step to verify the protection device library attributes and power details defined in the library, and satisfy the domain crossing requirements defined in the specifications. Designers also need to verify the difference between the power specification and design consistency.

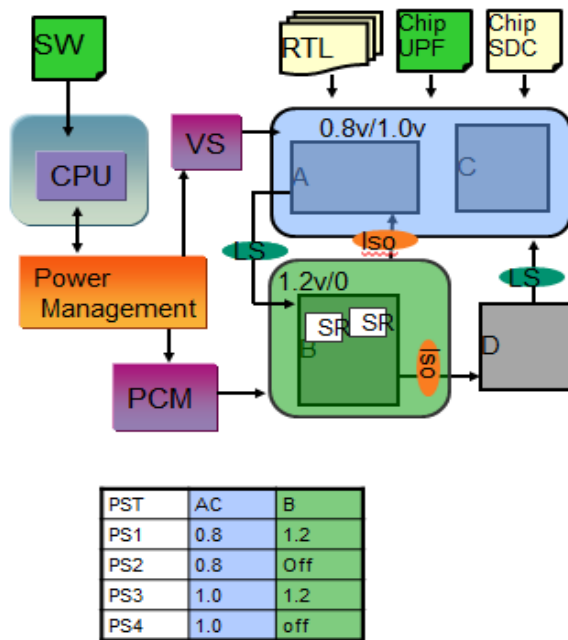
The ever increasing complexity of the SoC power architecture is forcing the specifications to have large number of power state tables (PSTs). Static tools employ certain merging principles to merge PSTs. Inconsistent or incomplete specification of the PSTs can lead to a merged PST which may not satisfy the actual low power intent. This may lead to incorrect verification results. Dynamic simulation techniques and manual debugging

of multiple PSTs can be very time consuming and inaccurate. Static tools can help to check inconsistency among the given PSTs by combining multiple PSTs into a single merged PST.

The user-defined power net specification may lead to some of the power related issues like leakage and over consumption (for example, the supply net associated to the node is ON for some multi-voltage state of legal state table, causing leakage in the path). Similarly, a node driven by an ON domain can cause the over consumption of power if it drives a node that is OFF. Some of these problems can be detected quickly by the static tools in the early stages of the design.

This paper describes a methodology using which you can combine the static and dynamic checks to successfully address the above low power issues in the early stages of the design. This minimizes the risk of subtle bugs escaping into silicon.

II. POWER MANAGEMENT VERIFICATION REQUIREMENTS



Low Power verification requirements can be listed as below:

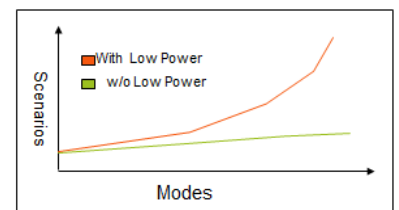
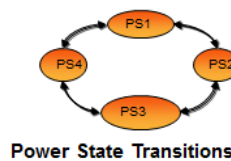
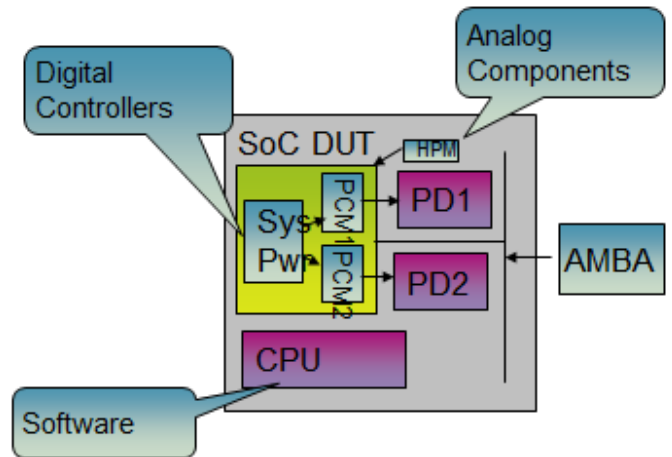
- **Verify the Power Control Management**
 - Ensure power transition when expected

- HW conditions that can cause power state changes
- Changes requested by SW
- **Verify the domains behavior in each state**
 - Check LP related behavior
 - Consider LP in normal checks
- **Coverage**
 - Exercise all states, all transitions, in all orders

III. LOW POWER VERIFICATION CHALLENGES

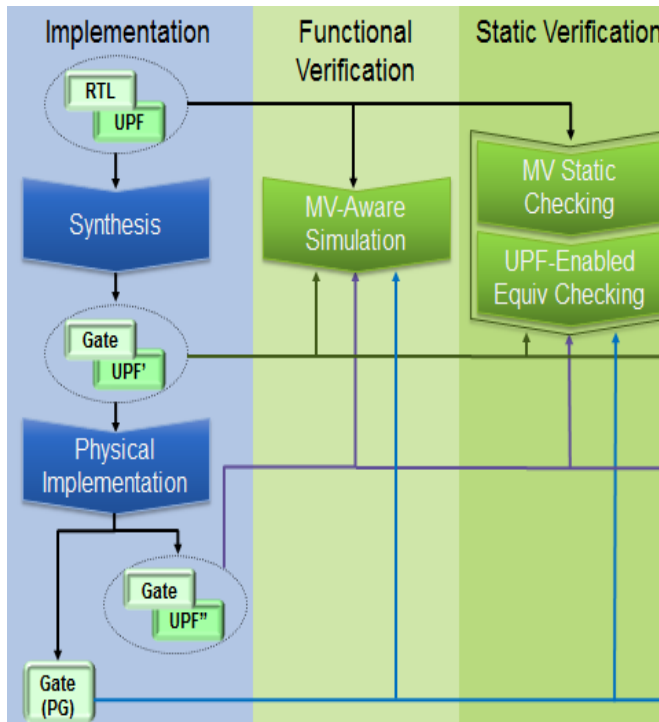
Low Power verification poses many challenges to the design and verification engineers. Some of these are listed below:

- **Huge verification space**
 - Large number of power states
 - Large number of transitions
 - Software applications
 - Firmware
 - Digital Hardware
- **System level verification**
 - Reuse in larger system
 - Often requires HW/SW simulations
- **LP specification rich in details**
 - And keeps changing



For tackling these verification requirements and challenges, there is a need for a methodology which ensures that all the low power verification goals are met accurately and in a timely manner.

IV. METHODOLOGY



The above diagram shows this methodology of combining static and dynamic low power verification.

The static verification can be used to clean the UPF before one can run dynamic simulation since it does not require a test bench. Once we have a clean UPF, dynamic simulation can then be run and power sequences can then be verified. Once the synthesis tool spits out a netlist, the static verification can then be applied (since running dynamic simulations on a netlist is both time consuming and performance heavy). Similarly, for a PG netlist, static verification could be used to verify the low power functionality.

While some people may argue for a need for combining dynamic and static verification, the fact is that each is incomplete in its own way. The highlights and pros and cons of dynamic and static low power verification are tabulated below:

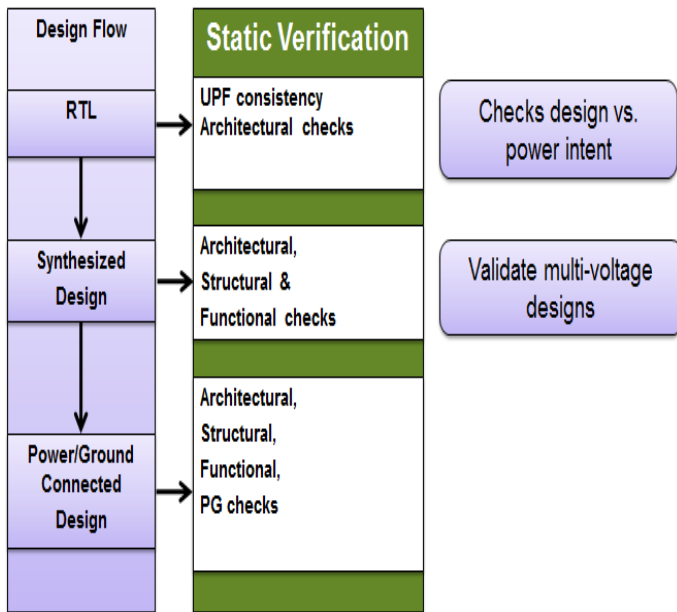
Low Power Dynamic vs. Static verification

Dynamic Verification
Main focus is on functionally correct power related "sequences"
The abstraction level is mostly RTL (GLS is both time consuming and has a huge performance impact)
Requires creation of test stimulus (verification can not start unless a power aware test-bench is in place)
Dynamic verification is needed to validate that the output values match with the expected values (e.g. if isolation clamp value is specified as '1', the isolation port should propagate '1'. Similarly the simulation output should show the correct value that needed to be retained on power up)
Static Verification
Main focus is on design related low power checks
This is very useful in validating RTL netlist or PG netlist
This can be very useful in flow-flushing UPF related issues (does not require a test-bench)
Static verification can not perform such checks

Performing gate level simulations for low power is a very slow and painful process. As a result the use of gate level simulations to verify low power has greatly reduced. But the question arises is how can we ensure the UPF consistency after the synthesis and place and route stages?

The answer is to perform the low power static checks throughout the design flow.

The below diagram illustrates this:



Thus for ensuring the low power verification completeness, we cannot just rely on either static or dynamic verification techniques.

Combining these would ensure that no subtle bugs escape into silicon.

V. EXAMPLES

What is PST: Power State Table (PST) is a construct in UPF that captures the legal combinations of power states for a set of supply ports/nets.

Ex : create_supply_port P1.

Add states to the Port

```
add_port_state P1 -state {HV 1.0} \
                    -state {LV 0.5} \
                    -state {OFF off}
```

Similarly for P2, P3 and P4

Add different states of the PST

```
add_pst_state s1 -pst PST_1 -state { HV HV LV }
add_pst_state s2 -pst PST_1 -state { HV HV LV }
add_pst_state s3 -pst PST_1 -state { HV LV off }
```

Why PST is important:

The number of combinations of all port states can be large.

If power ports P1, P2, P3, P4 contains three port states each, then the number of combinations can be $3^4 = 81$. The PST gives a directive to the tools that among the 81 states, only 3 are legal. The rest are all illegal states.

Synthesis tools rely solely on PSTs for Level Shifter insertion.

PST is the only specification to give legal voltage values.

What is PST Merging?

In hierarchical UPF flow, there can be multiple UPFs and multiple PSTs defined at different scopes. At the chip top scope static tool creates a merged PST which contains all the possible combinations of all the scoped PSTs states. Static solution checks the common states in all the PSTs. If there are common states, the tool overlaps the PSTs and derives the intersection of all the PSTs. This derived PST is the final PST that takes effect.

Different block/scope level UPFs have different PSTs defined at their block/scope level. There might be chance that different block works at different voltages. When those blocks are used at the SOC level, Static solution needs a golden PST which contains all the possible states of all the blocks and creates a merged PST. On that common merged PST, Static checkers does further low power analysis.



Merged PST

	TOP	BLK A	BLK B
merged_pst_s t0	1.2	1.2	1.2
merged_pst_s t1	0.8	0.8	0.8

Advantage: When you have large SOC and many IP getting integrated each has its own UPF. PST size will be huge and problems can be multiple. PST Merge will uncover many issues at the early stages of the design. This static LP feature will help towards smooth implementation flow.

2] Dynamic voltage scaling is the power-management technique where primary voltage is varied within a certain range of active voltage levels to achieve different performance and power operating points. For example, in a low performance mode voltage is lowered to achieve lower leakage. In a high performance mode, voltage is increased to achieve higher performance but

at the cost of higher leakage consumption. It is a trade-off dynamically chosen by the power-controller of the chip to achieve the optimal power-performance parity.

In a design using DVS, following are the key impacts on the power-domain boundaries that the simulation engine must comprehend. For each signal going from one domain to another the simulation must,

- identify if any level-shifting is required
- infer level-shifters as necessary based on information provided in power-intent (UPF)
- enable voltage level aware simulation semantic if level-shifters are absent. All these would add at least 5-6x performance hit on the simulator. The best preferred way to explore, all the different Level shifter as well as PST checks which is much faster and quicker.

3] These days SOC's designers add protection devices in the RTL itself and its supply connections are define in UPF through the Connect supply net and Strategy supply and domain supply e.tc. This can be used by the implementation tools to build the supply network.

There can be many issues which can be detected early in the design flow as follows.

Electrical violations caused by incorrect supply specifications for protection [ISO/LS/ELS] cells.

Iso rail order violations: Isolation required but isolation rail is OFF when the destination is ON.

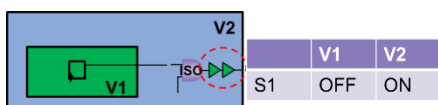
Solution is Fix the supply connections or PST supply. SRC and DEST states in PST determine need for protection

Protection cells are inserted to fix above violations Protection cells themselves must NOT cause additional violations in all PST states

Rails to protection cells or PST must be corrected to fix any further violations introduced by protection cells.

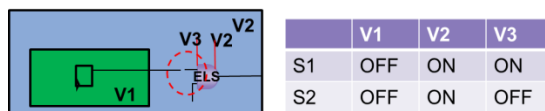
Rail order violation : corruption

A node driven by an OFF rail is driving the a node that is ON and the corruption is able to reaches destination (i.e., no isolation beyond the OFF node)

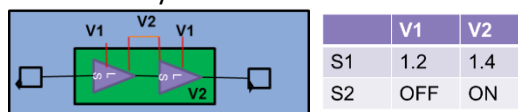


Rail order violation : Leakage

A node driven by an OFF rail is driving a node that is ON but the corruption does NOT reach the eventual destination (i.e., there is isolation beyond OFF node)

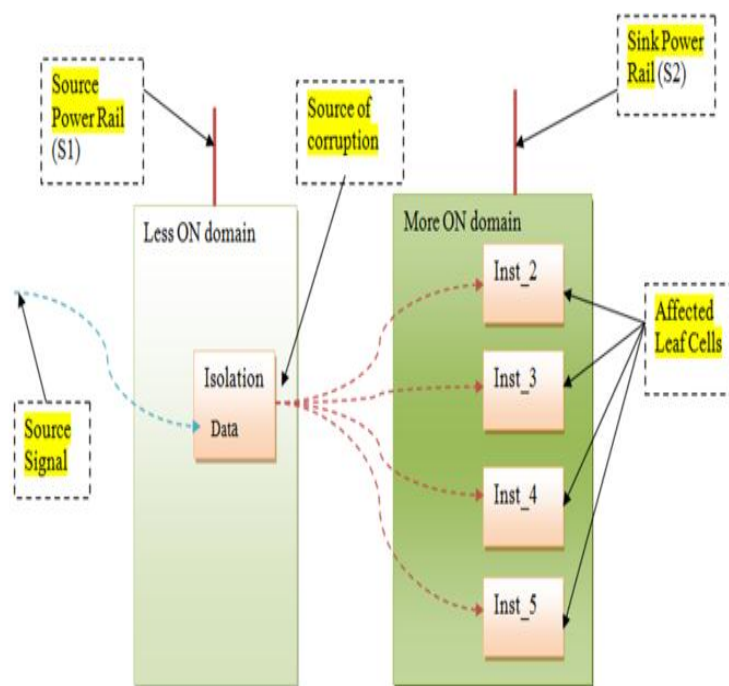


A node driven by an ON rail is driving a node that is OFF unnecessarily



5] Global signals and Control signals like Clock Reset, Scan Enable, Iso_enable, Switch Enable, Save Restore shouldn't go from lower order Power domain to higher order power domain. Because when source is OFF, signal can't reach destination domain.

When the protection devices present on these signals Corruption happens due to isolation.

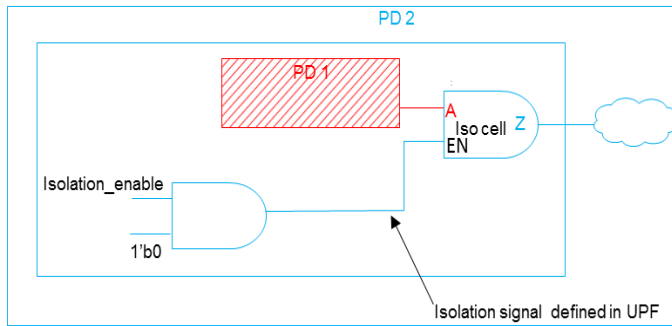


These issues can be detected if the test scenarios are covered in dynamic LP verification if the test environment cover the functionality but can be detected easily and quickly in Static checks.

6] If there is a combinational logic on the enable path if one of the input of the combinational logic is tied by constant then these constant can be dominating such

that either the protection device is in isolated mode else the enable will not reach the isolation gate.

In dynamic simulations these can be detected if there is a specific tests scenario. These checks need to be performed on the all the protection devices would add a lot of test overhead. Static solution will detect the combo gate and to backward extraction and if there is a constant it can propagate and check what is the effect on the constant on the enable. This is much faster and quicker.



VI. CONCLUSIONS AND FURTHER DEVELOPMENTS

The combined dynamic and static low power verification flow provides a full proof verification. It ensures that no bugs in a low power design escape silicon.

The following can thus be concluded based on this flow:

- Dynamic simulation checks for the correctness of low power sequences and the correct data values upon power wake up
- Static verification checks for the structural and architectural low power checks
- Each flow is incomplete in its own way so combining them ensures the low power verification completeness

As a part of future development, another addition to this flow which is being worked upon is the equivalence checking flow at each stage i.e. rtl, gate level netlist and PG netlist.

VII. ACKNOWLEDGMENTS

The authors would like to thank the entire Low Power dynamic and static verification team without whose support this flow would not have been a success.

VIII. REFERENCES

- [1] IEEE Standard for Design and Verification of Low Power Integrated Circuits