“Shift left” Hierarchical Low-Power Static Verification Using SAM

Bharani Ellore bharani.ellore@amd.com (US, Austin)
Parag Mandrekar parag.mandrekar@amd.com (US, San Jose)
Himanshu Bhatt himanb@synopsys.com (US, Mt. View)
Susantha Wijesekara susantha@synopsys.com (US, Mt. View)
Bhaskar Pal bpal@synopsys.com (India, Bangalore)

ABSTRACT

With increasing SoC complexity, growing design sizes and advanced power-aware architectures, early and efficient static low power verification is important to reduce turnaround times and enable faster time to market. For hierarchical verification, designers use a black box flow, Liberty Model based hierarchical flow, ETM flow or a glass box flow that offer various degrees of trade-offs between accuracy and performance. While the black box flow can be best for performance, the full flat run may give better quality of results. The Synopsys VC LP solution has a new flow with Signoff Abstract Models (SAM) for hierarchical verification, which is designed to provides the same QoR and achieves better performance than flat runs. The paper showcases these methodologies along with the results that can be achieved with a “shift-left” in overall low power static verification signoff.

1. Introduction

With increasing SoC complexity, growing design sizes and advanced power-aware architectures, early and efficient static low power verification is important to reduce turnaround times and enable faster time to market.

Growing design sizes, low power (LP) complexity and the need for early stage verification is making designers adopt hierarchical verification flows. Traditionally for hierarchical verification, designers use a black box, Liberty Model based hierarchical flow, timing model (ETM) flow or stub/glass box flows that offer various degrees of trade-offs for accuracy of the results and performance. While black box flow is best for performance, full flat runs give the best quality of results as full design is available for checks. Adopting a new flow, Signoff Abstract Model (SAM), for hierarchical low power verification can provide guaranteed QoR by retaining enough logic at the sub-module level to deliver much better runtime performance at the SoC level compared to flat runs. Additionally, this flow enables the SoC integrator to focus on top level violations and integration related issues and not worry about violations deep inside the hierarchical blocks, since the block owners would sign-off their blocks after review of the violations. With this efficient solution, there is runtime performance gain and reduced memory consumption as compared to the full flat verification, while not losing any QoR, and greatly reducing reduce the turnaround time (TAT) during low power verification sign-off.

2. TRADITIONAL HIERARCHICAL VERIFICATION FLOWS VS. NEW TECHNOLOGY (SAM)

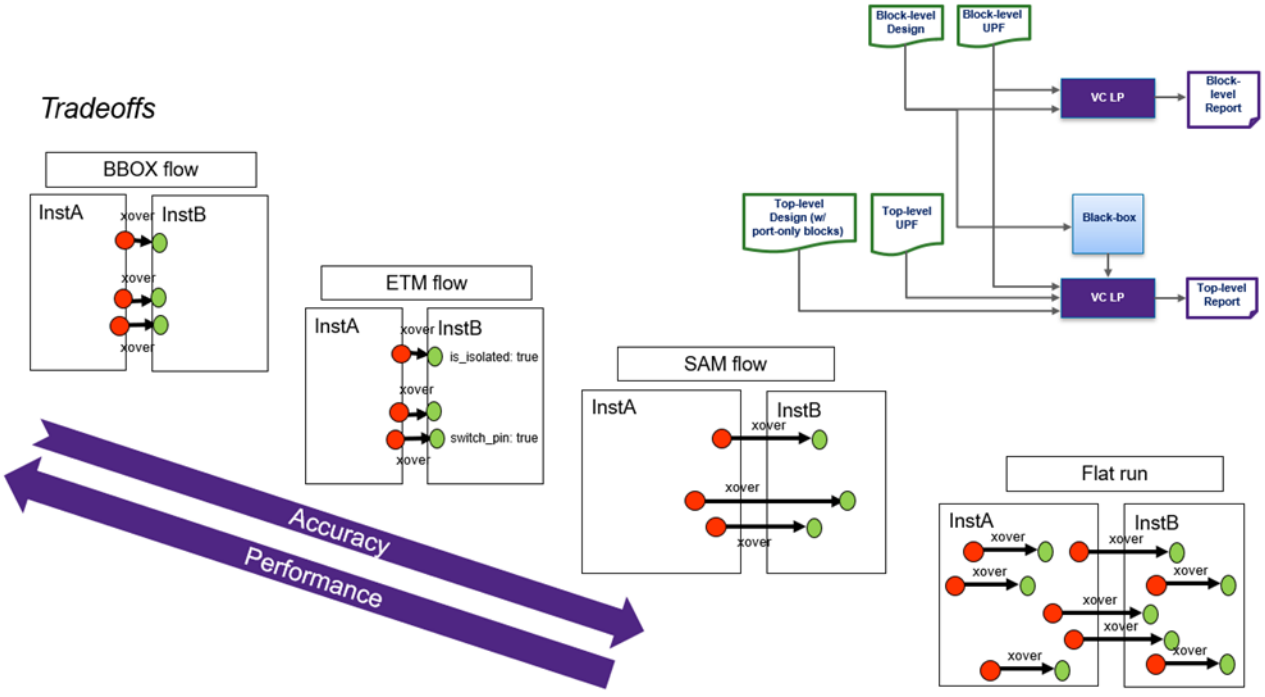

Figure 1: Comparison of hierarchical low power verification flows

- In traditional approaches, partitions are completely black boxed – signoff is not guaranteed.

- The new flow generates a boundary accurate SAM model for the partitions.
- Guaranteed QoR – proven with additive and subtractive QoR flows.
- Up to 15x performance seen on netlist designs.

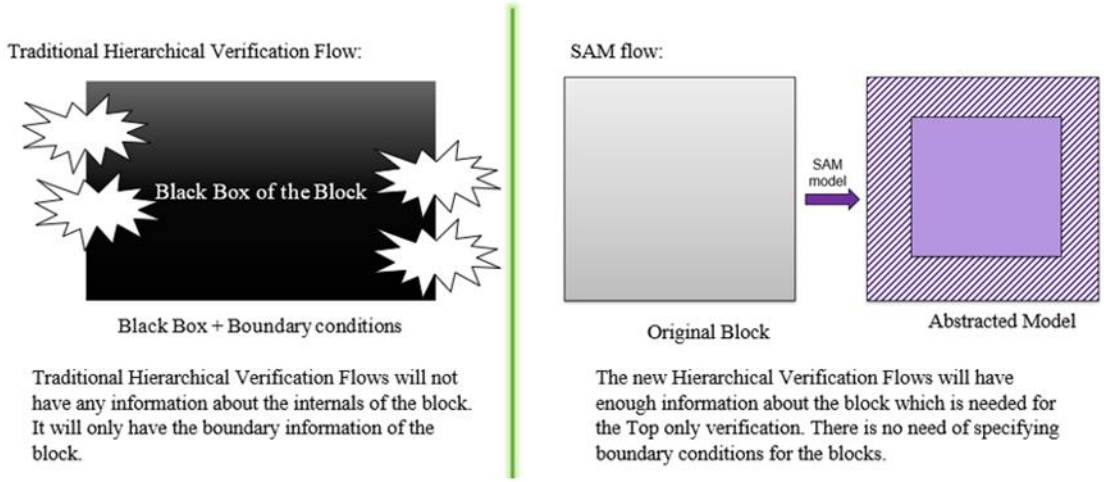

Figure 2: Black Box flow vs SAM flow

2.1 LOW POWER HIERARCHICAL VERIFICATION METHODOLOGY

In this new hierarchical flow, blocks that are integrated into the SoC will need to be abstracted first using a low power static checker, such as Synopsys' VC LP solution. During abstraction, the hierarchical instances and net connections that are not needed for top verification activities are removed and an abstracted model will be dumped into a new HDL file. This HDL model can be loaded into the SoC instead of the original block to perform the SoC verification. The benefits of using this flow includes less memory usage, improved run time and focused violations (violations reported will be mainly related to the top-level integration).

The following diagram shows how hierarchical designs can be abstracted and used in hierarchical blocks to achieve the same QoR as flat runs but with improved performance and reduced turnaround time.

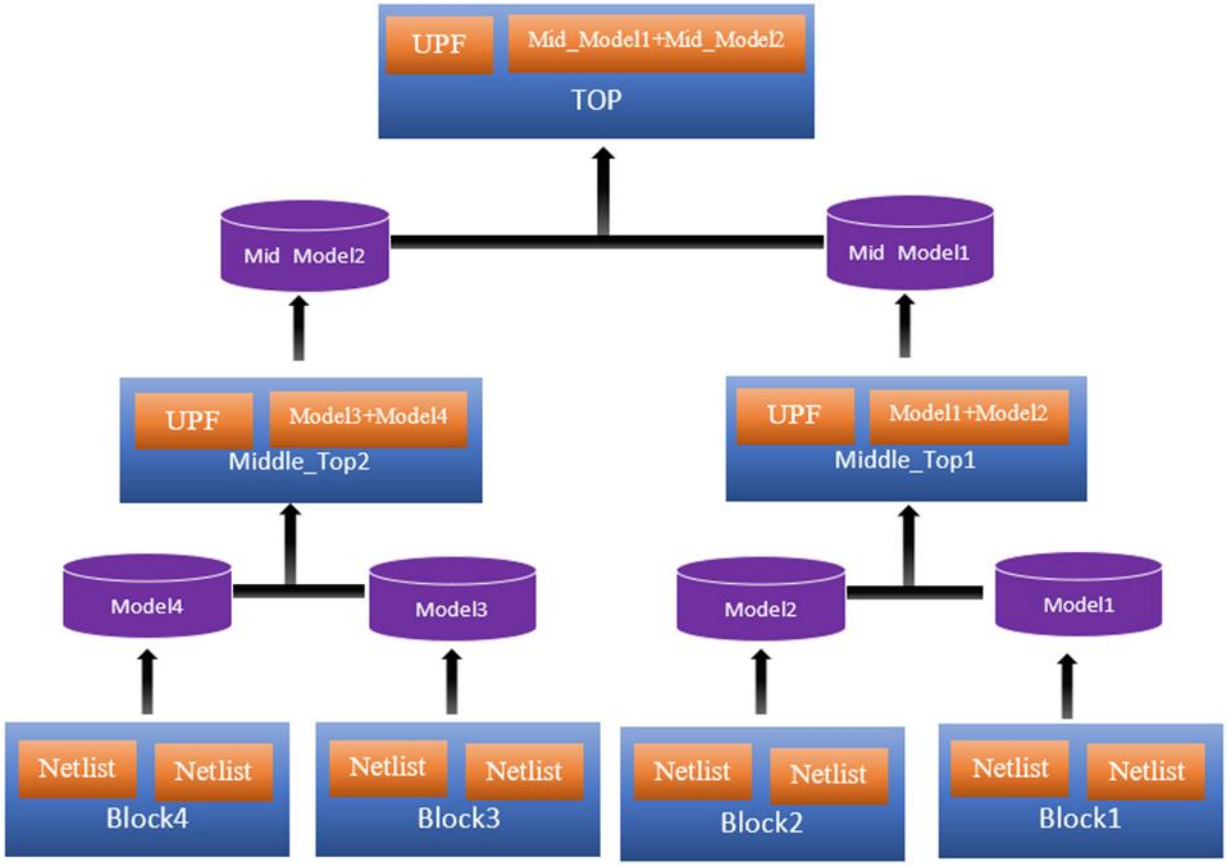

Figure 3: SAM Hierarchical flow

2.2 CHARACTERISTICS OF THE ABSTRACT MODEL

The abstract model is generated to contain the minimal set of logic needed for top level verification, which provides a lightweight model compared to the original full block netlist. This will help improve the runtime of the static checking tool during design read, as well as LP checking.

During abstraction, the tool will model necessary logic for complete verification. As a result, the designer does not need to specify any special constraints on the boundary ports of the blocks. Since all the required logic is modeled there are no missing violations during the SoC verification.

2.3 Creating/Writing a SAM Model

source synopsys_vcst.setup
set search_path "."
set link_library “a.db b.db”
configure_lp_abstraction
read_file -verilog -netlist BlockA.vg -**top BlockA**
read_upf ./src/BlockA_gate.upf
mark_lp_abstraction –netlist|-pgnetlist
write_verilog_abstract_model -path foo/bar
report_lp
quit

2.4 Reading a SAM Model

source synopsys_vcst.setup
set search_path "."
set link_library “a.db b.db”
configure_lp_tag –tag * -enable //enable your rule set
set_verilog_abstract_model –module BlockA
set_verilog_abstract_model –module BlockB
read_file -verilog -netlist “
foo/bar/BlockA/Verilog/BlockA_abstract.v
foo/bar/BlockB/Verilog/BlockB_abstract.v
top.vg”
-top
load_upf ./src/top.upf
infer_source
foreach f [glob [runtime_db]/reports/*.tcl] {
  source $f
}
check_lp -stage {upf design pg}
report_lp
quit

2.5 VALIDATION OF THE RESULTS

The new hierarchical verification flow will ensure that there will be no loss in coverage of LP violations. All the violations reported in top level flat run will be reported in either block level flat run or Top+Model run.

2.6 RESULTS

The table below shows that the Static Abstraction Model (SAM) based hierarchical flow provided QoR the same as flat run in conjunction with improved performance both in terms of reduced runtime and runtime memory.

Design	Top Flat Runtime(hrs/mins)	Top+SAM Runtime(hrs/mins)	Runtime Gain	Top Flat Memory (MB)	Top+SAM Memory (MB)	Mem Gain
Design 1	28hrs	8hrs 49mins	3.17X	792926	239527	3.3X

3. Conclusions

Next-generation SoCs with advanced graphics, computing, machine learning and artificial intelligence capabilities are posing new unseen challenges in low power verification. Traditional LP hierarchical verification flows are not scaling up for new designs that are being taped out with billions of transistors and large number of power domains. Static Low Power Verification tools using hierarchical verification technologies, enable a "shift-left" in the overall verification TAT and at the same time ensure that there is no loss of QoR.

4. References

[1] 1801-2015 - IEEE Standard for Design and Verification of Low-Power, Energy- Aware Electronic Systems

[2] https://semiengineering.com/efficient-hierarchical-verification-for-low-power-designs/

[3] https://www.techdesignforums.com/practice/technique/low-power-debugging-made-easy/