Sub-design Interface Aware Top Only Static Low Power Verification

1"Heichang Lee, Nikhil Amin",
Mail address: 5F. H-Square N-dong, 235, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13494, Korea
Email: {helee, Nikhil.Amin}@synopsys.com
2"Jianfeng Liu, Minyoung Mo, Dongkwan Han",
Mail address: DSR-C 12th floor, 1-1 Samsungjeonja-ro, Hwaseong-si, Gyeonggi-do, Korea
Email: {jf.liu, my0911.mo, dk.han}@Samsung.com

***Abstract*- Discontinuity between block level verification and top level verification is a major cost in low power verification. There has been no methodology of conducting compelling top level verification unless block level designs are complete. Full chip check guarantee the best check quality but slow and tedious. Existing top only flows are faster but check quality is not comparable to full chip. This paper proposes sub-design interface aware top only static low power check to overcome time and quality of existing methodologies. The new flow ensures consistency of the top level check results when sub-design has golden power intent and proper supply constraints.**

I. INTRODUCTION

As SoC has required hierarchical implementation and verification approaches, a static low power check has been conducted at two phases: block level check and top level check. The easiest and the most inefficient way of completing the verification is to complete all block level designs, IPs, and macros first and then, complete the top level verification after putting all blocks together [1]. This approach not only costs time but also causes redundant block level violations reproduced at the top level.

Another existing approach is the top only verification flow by manually black-boxing block designs or replacing sub-designs with Extracted Timing Models (ETM) [2]. The key advantage of this flow is to be able to perform early top level check before sub-designs are complete. However, manual black-boxing requires coding for wrappers and Unified Power Format (UPF) for black boxes. Handling separate UPFs for black boxes makes it harder to manage consistent UPFs through a project.

When it comes to ETM flow, collecting all ETMs from implementation results in time and communication overhead between verification and implementation teams. In addition to the ETM generation overhead, captured power intents in ETM precede any other additional power intent. In other words, it is hard to put user intended supply constraints after ETM is generated. In the top only flow, it is important to force user intended supply constraints at the sub-design interface ports for accurate static low power verification.

A new low power verification methodology is proposed in this paper to provide seamless connection between block and full chip verification by enabling early stage top level verification whose check quality is comparable to that of the full chip. The design under test is a mobile application processor (AP) SoC from a chip maker. The AP has nearly thirty instances of sub-designs. Each instance has several power domains. The number of sub-UPFs are the same as the number of sub-designs. The static rule checker, VC-STATIC-LP (VC-LP) 2015.09-SP2-2, is used for the evaluation of this new methodology.

II. APPROACH

This paper demonstrates Sub-design (block) interface aware top only verification flow and compares the results of the new flow and the full chip check flow. The prerequisite for the flow is the top design, sub-designs at any stage (either complete, incomplete, or port wrapper), golden UPFs for the top and sub-designs and user intended supply constraints.

This flow automatically preserves block boundary power intents and block level port information. The static rule checker, VC-LP, supports the flow without requiring extra overhead in user’s VC-LP script [3]. As it wipes off everything inside a block interface, top level check can be done at any stage during the project time line, regardless of the progress of the block level implementation.

A. *Sub design interface information*

A block design information includes Verilog netlists, port attributes extracted from libraries of cells and IPs. Sub-design netlists comprise logic modules, cells and ports with directions. In this paper, the sub-design interface aware flow only requires boundary ports and attributes of a design. The rule checker should be aware of a port direction and whether a port is a supply port or a logic port as described in Fig 1. In case of a macro IP, its port attribute is obtained from the given macro library.

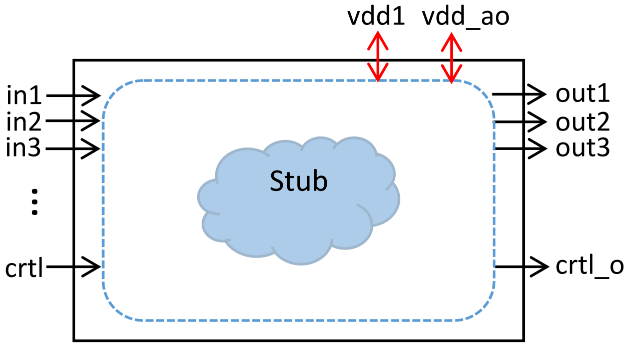
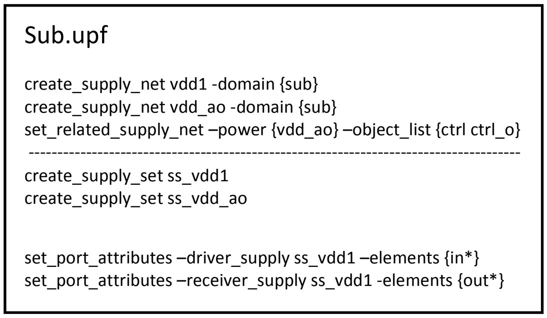


Figure 1. Boundary supply/logic port.

A sub-design has its own power intent described with UPF. A power intent consists of power distribution networks, power strategy, power modes, and mapping of LP cells. A power distribution network defines power supplies used in the design, primary supplies and supply constraints at every interface port. A supply must be designated at the interface port for correct supply relationship between the block boundary and the top logic. Fig 2 shows a sample UPF and Fig. 3 visualizes the supply relationship of Fig. 2.



```
Sub.upf

create_supply_net vdd1 -domain {sub}
create_supply_net vdd_ao -domain {sub}
set_related_supply_net –power {vdd_ao} –object_list {ctrl ctrl_o}
------------------------------------------------------------------------------------
create_supply_set ss_vdd1
create_supply_set ss_vdd_ao


set_port_attributes –driver_supply ss_vdd1 –elements {in*}
set_port_attributes –receiver_supply ss_vdd1 -elements {out*}
```

Figure 2. Related supplies at interface ports in UPF.

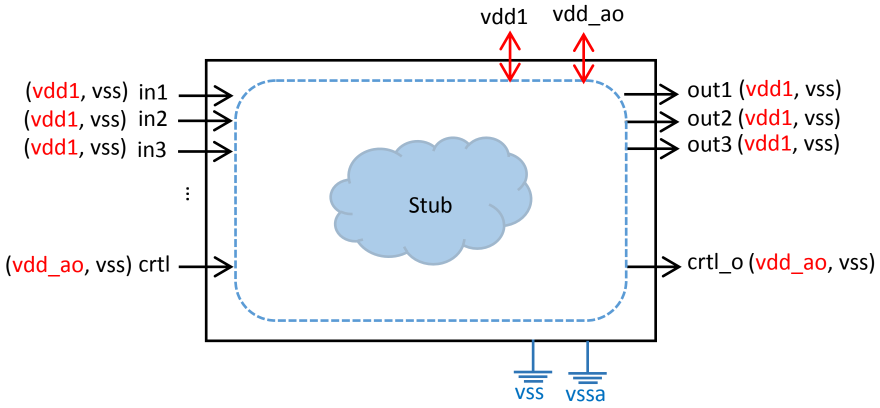
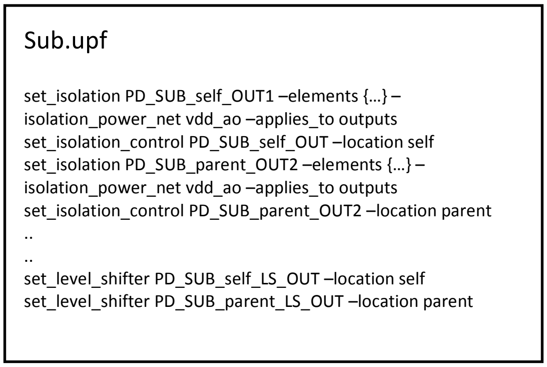


Figure 3. Related supplies at interface ports.

Power strategies in a UPF define isolation, level shifter, and power switch rules including their control signals. Fig. 4 states target locations of isolation and level shifter strategies. Depending on sub blocks, isolation strategies are placed either at fanout side or internal side of the design interface. The UPF in this paper uses “self” or “parent” location. The level shifter positions are described in the same manner. All strategies inside sub-designs are discarded by VC-LP. All implemented cells inside sub-designs are also ignored by VC-LP. Fig. 5 shows how sample “sub.upf” is handled.



Sub.upf

```
set_isolation PD_SUB_self_OUT1 –elements {…} –
isolation_power_net vdd_ao –applies_to outputs
set_isolation_control PD_SUB_self_OUT –location self
set_isolation PD_SUB_parent_OUT2 –elements {…} –
isolation_power_net vdd_ao –applies_to outputs
set_isolation_control PD_SUB_parent_OUT2 –location parent
..
..
set_level_shifter PD_SUB_self_LS_OUT –location self
set_level_shifter PD_SUB_parent_LS_OUT –location parent
```

Figure 4. Strategies of isolation and level shifter in UPF.

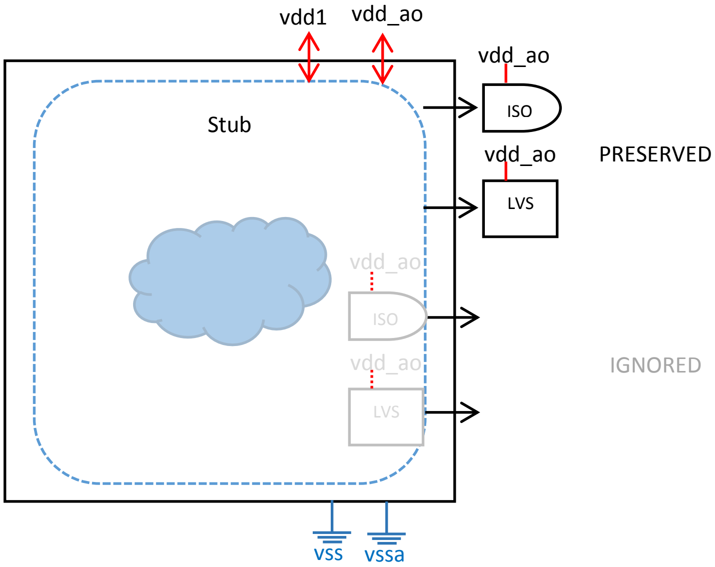
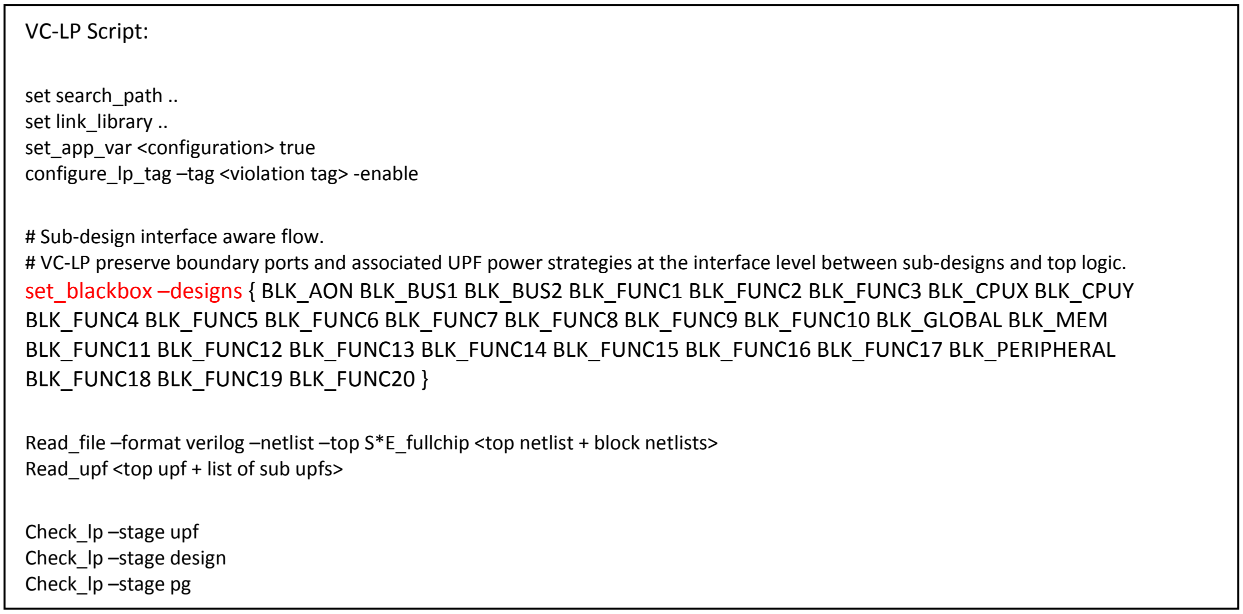


Figure 5. UPF handling for a sub-design by VC-LP.

B. Methodology

The proposed methodology takes two major steps: reading designs, and checking designs. Fig. 6 is the sample VC-LP script for the methodology. During the reading design stage, VC-LP reads in whole designs and power intents including top and block level netlists and UPFs.

VC-LP script must assign which sub-design to be black boxed with “set_blackbox -designs” command. User can choose any sub-designs in the “set_blackbox” command.

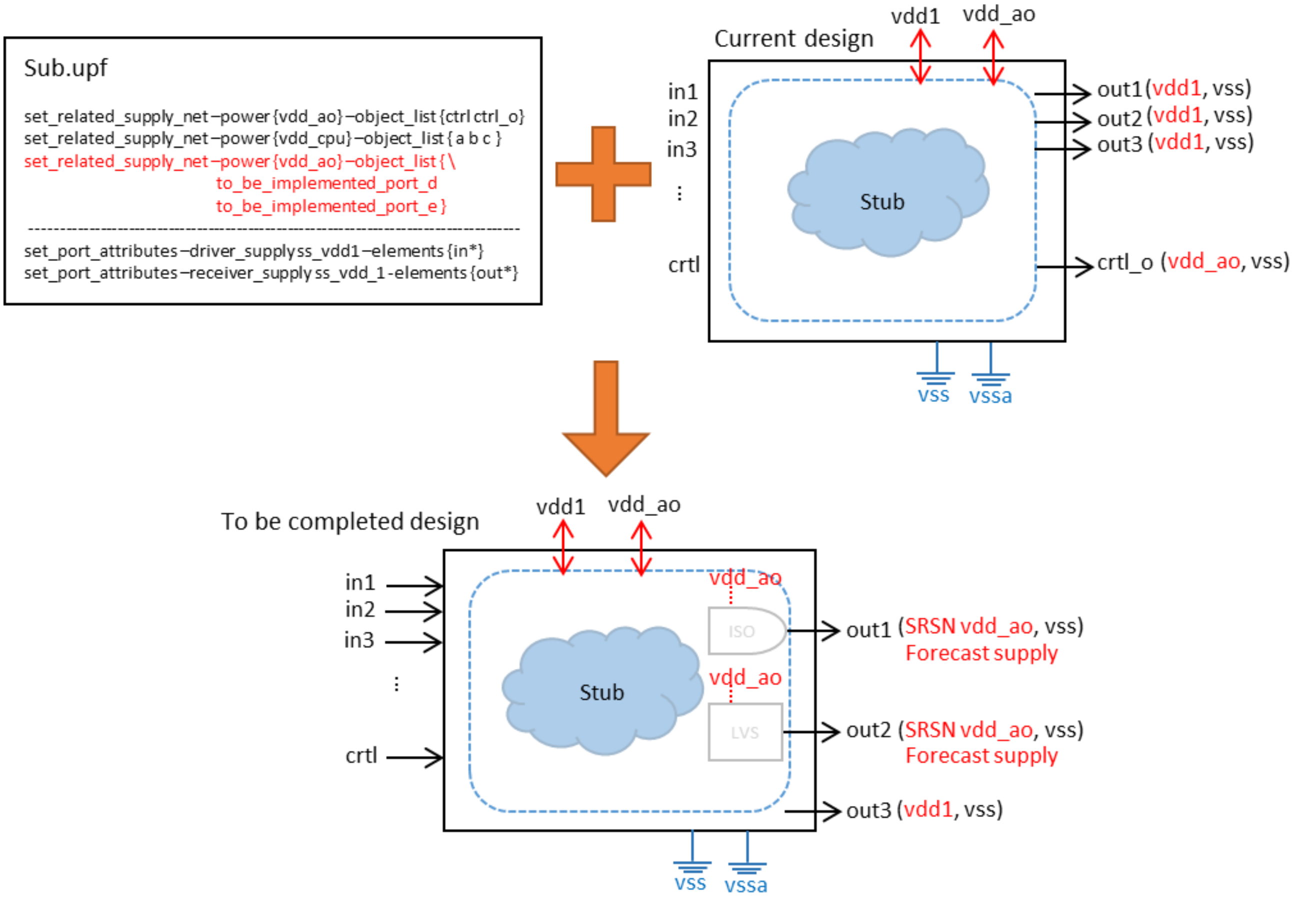


```
VC-LP Script:

set search_path ..
set link_library ..
set_app_var <configuration> true
configure_lp_tag –tag <violation tag> -enable

# Sub-design interface aware flow.
# VC-LP preserve boundary ports and associated UPF power strategies at the interface level between sub-designs and top logic.
set_blackbox –designs { BLK_AON BLK_BUS1 BLK_BUS2 BLK_FUNC1 BLK_FUNC2 BLK_FUNC3 BLK_CPUX BLK_CPUY
BLK_FUNC4 BLK_FUNC5 BLK_FUNC6 BLK_FUNC7 BLK_FUNC8 BLK_FUNC9 BLK_FUNC10 BLK_GLOBAL BLK_MEM
BLK_FUNC11 BLK_FUNC12 BLK_FUNC13 BLK_FUNC14 BLK_FUNC15 BLK_FUNC16 BLK_FUNC17 BLK_PERIPHERAL
BLK_FUNC18 BLK_FUNC19 BLK_FUNC20 }

Read_file –format verilog –netlist –top S*E_fullchip <top netlist + block netlists>
Read_upf <top upf + list of sub upfs>

Check_lp –stage upf
Check_lp –stage design
Check_lp –stage pg
```

Figure 6. Sample VC-LP script for the sub-design interface aware top only rule check.

With read_file and read_upf commands, the static verification tool starts to parse and drop designs and power intents inside a block design boundary. Power intents at block design interfaces are sustained unless the power strategy written to place a cell inside the design.

Sub-design UPFs must include SRSN/SPA to represent internal supply or forecast supply of the boundary port. User can assign SRSN that will match the supply of completed sub-designs. Fig 7 shows the impact of SRSN in the sub-design UPF.



Sub.upf

```
set_related_supply_net –power {vdd_ao} –object_list {ctrl ctrl_o}
set_related_supply_net –power {vdd_cpu} –object_list { a b c }
set_related_supply_net –power {vdd_ao} –object_list { \
                                  to_be_implemented_port_d
                                  to_be_implemented_port_e }
--------------------------------------------------------------------------------
set_port_attributes –driver_supply ss_vdd1 –elements {in*}
set_port_attributes –receiver_supply ss_vdd_1 -elements {out*}
```

Figure. 7 Related supplies when forecast SRSN are applied.

Preserving sub-design interface power intents enables our methodology to conduct low power rule checks between sub-design interfaces and the top logic because the interface power intents can represent internal supply.

Fig. 8 displays the final form of the design and the UPF interpretation for sub-design interface aware top only static check.

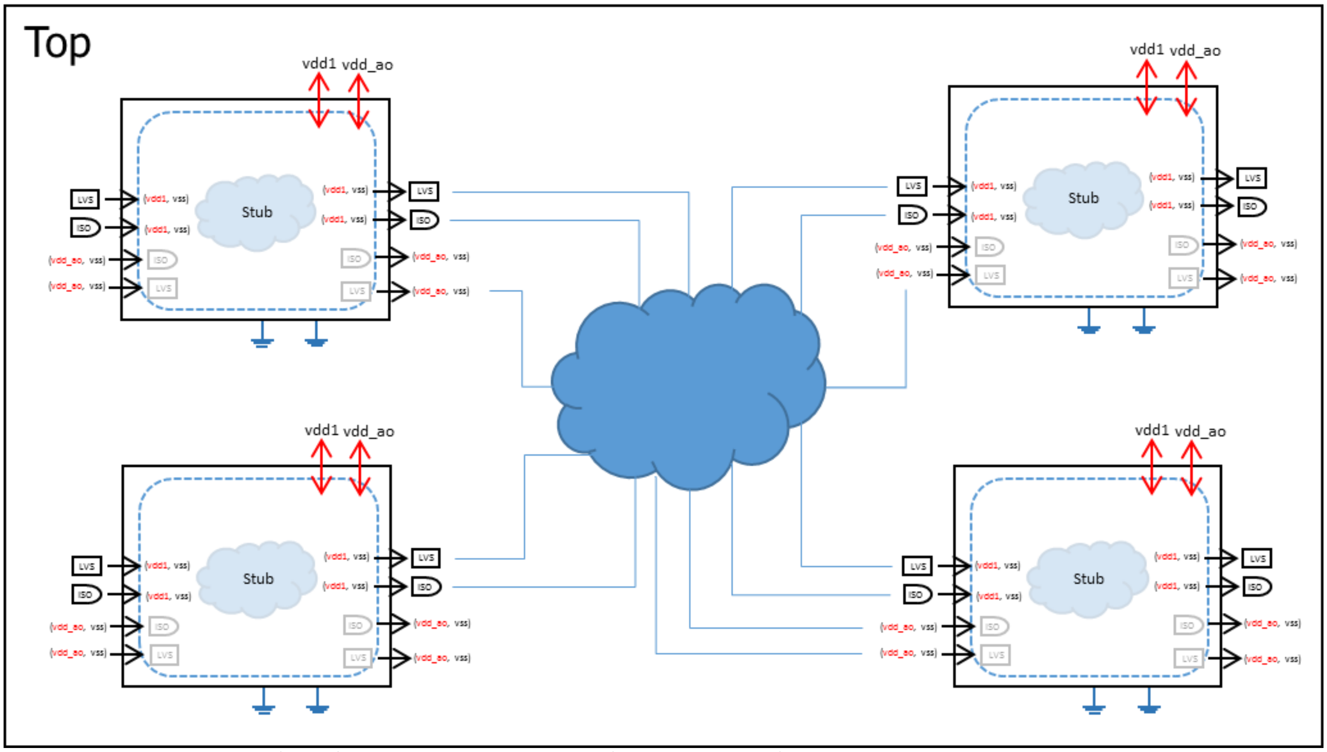


Figure. 8 Final form of design and UPF interpretation for sub-design interface aware top only static check.

Checking design stage conducts UPF to design consistency check, design structural and architectural checks. Due to black boxing of the internal designs and UPFs, our methodology is not able to detect violations inside sub-designs. In terms of top level rule checks, results between full chip and the block interface aware flow are consistent if interface ports are constrained with expected supplies by verification engineer.

III. Rule Check Quality

A. Processing sub-design UPFs

We have examined the static checker messages during UPF read stage. Table 1 lists out messages regarding dropping internal UPF descriptions of sub-designs. Only internal UPF commands and strategies are ignored as expected. Sub-design interface strategies and supply constraints are preserved. If no supply constraint such as 'set_related_supply_net' (SRSN) or 'set_port_attributes' (SPA) are present for an interface port, the tool warns with "no supply constraint". By default, an unconstrained port takes the primary supply of the sub-design. In case of an unconstrained port that must relate to a non-primary supply, it will lead to incorrect rule check results. *The most important part of the sub-design UPF is to specify SRSN or SPA to express expected internal supply of the sub-design.* Only then, rule check will be performed with the correct supply relationship between top and sub-designs.

TABLE I
VC-LP UPF HANDLING FOR SUB DESIGNS

UPF handling	**Description**
Remove internal ISO/LVS	ISO/LVS strategy with 'self' location removed.
	ISO strategy including Blackbox (sub-design) elements ignored.
Remove internal UPF commands and supply network	UPF Object ignored in Blackbox UPF

	UPF Command ignored in Blackbox UPF (create_power_domain, set_retention, …)
Warn supply constraint of a boundary port	Overriding the related power for design port “port_a[0]” …
	Sub-design port has no supply constraint

B. *UPF violations comparison*

This section highlights the rule check results with respect to UPFs between full chip and sub-design interface aware top only flow. Table 2 summarizes identical violations between both flows. PST_STATE_MULTIPLE [4] indicates that multiple same power modes exist. UPF_CSN_LS means some of the level shifter power pins are not assigned to a supply. UPF_CSN_PAD is like UPF_CSN_LS but is for pad supply pins. UPF_SUPPLY_MISSING warns that a declared supply is not included in the power state table which lists all power scenarios. PST_SUPPLY_MULTIPLE points that the same supply appeared multiple times in the power state table which could contradicts its operation. The common point of these violations is that UPFs are written against the top design components.

TABLE 2
CONSISTENT RESULTS AT UPF CHECKS

Violation Tag	Counts	Description
PST_STATE_MULTIPLE	134	Same power state
UPF_CSN_LS	2088	LS pg pin requires CSN.
UPF_CSN_PAD	1516	PAD pg pin requires CSN.
UPF_SUPPLY_MISSING	65	UPF supply never present in PST.
PST_SUPPLY_MULTIPLE	6	Same supply did not appear in the power state table.

Table 3 shows the inconsistent results of the two flows. Tags in Table 3 have majority of consistency between full chip and the sub-design interface aware flow but some mismatches are mainly due to a wrong golden upf or lack of supply constraints. We will discuss how to handle these mismatches with SRSN/SPA. Table 3 shows the number of unique cases instead of counts. This is because counts can be large for one unique root cause, depending on the chip instances, fanout sizes of the violation. It makes more sense to identify unique cases than just counts.

TABLE 3
ROOT CAUSE COUNT OF INCONSISTENT RESULTS AT UPF CHECKS

Violation Tag	Unique cases	Description
ISO_STRATEGY_MISSING	2	UPF is missing isolation strategy for a crossover
LS_STRATEGY_MISSING	1	UPF is missing level shifter strategy for a crossover
UPF_SUPPLY_NO_STATE	1	Created supply not registered in the PST
UPF_SUPPLY_UNCONN	1	UPF Supply does not have any driver
UPF_SRSNSUPPLY_MISMATCH	-	SRSN supply is not identical to driver/load supply/SRSN
UPF_SUPPLY_UNUSED	1	Declared supply not used anywhere in UPF

ISO_STRATEGY_MISSING mismatch is resulted from improperly defined SRSN on the sub-design interface ports. This design has placed isolation strategies and cells inside the domain wrapper boundary. When a sub-design is black boxed, internal isolations are lost. To overcome this situation, SRSN should express expected output supply (or forecast supply) of the isolation instance at the interface port as shown by the left side of Fig 9. If UPF takes advantage of a supply set, SPA receiver supply can be specified instead of SRSN as shown by the right side of Fig 9.

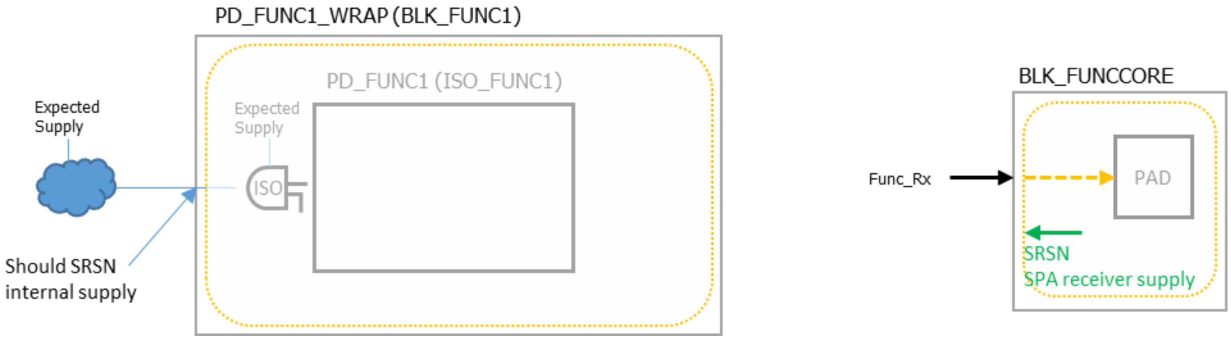


Figure. 9. Disconnected supply information at the crossover. SRSN/SPA handling to overcome.

LS_STRATEGY_MISSING is a UPF to design mapping issue on the full chip. As Fig. 10 described, the UPF and the design are correctly implemented. But the strategy was not mapped to the node. This violation did not appear in our top only flow.

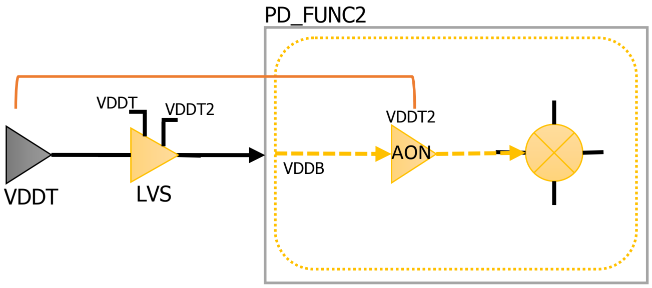


Figure 10. False alarm of LS strategy mapping in full chip.

UPF_SUPPLY_NO_STATE indicates that a block level supply does not have voltage and state defined. It only appears in the full chip flow because of the wrong golden UPF. UPF_SUPPLY_UNCONN from the full chip flow is due to a similar reason as of UPF_SUPPLY_NO_STATE.

UPF_SUPPLY_UNUSED is a limitation of the sub-design interface aware flow. Some of the block supplies are connected to internal cell power/ground (pg) pins. Internal connections can only be identified either by a block level verification or by a full chip check.

UPF_SRSNSUPPLY_MISMATCH is a conflict between source side SRSN/supply and sink side SRSN/supply. Verification engineers should make sure that SRSN of the source and the sink are intentional.

C. *Design violations comparison*

Design check focuses on structural checks. Structural check includes supply rail violation, Low Power (LP) cell position and its associated supplies. In Table 4, both RAIL_ELSOUTPUT_STATE and DESIGN_BACKUP_STATE were reported from the logics in the top level.

TABLE 4
CONSISTENT RESULTS BETWEEN SUB-DESIGN INTERFACE AWARE AND FULL CHIP.

Violation Tag	Counts	Description
RAIL_ELSOUTPUT_STATE	2	LVS output supply is OFF when sink is ON.
DESIGN_BACKUP_STATE	33	Backup power is OFF when primary power is ON.

Table 5 lists out inconsistency because of improper SRSN constraints and the limitation of the new flow. Critical violations such as ISO_INST_MISSING and LS_INST_MISSING can be resolved by adding SRSN/SPA on the sub-design interface if ISO/LS instances exist in the sub-design. Other minor violations are either because of the improper SRSN or due to the limitation of our methodology.

ISO_INST_MISSING is one of the most dangerous violations. Therefore, supply constraints on the boundary ports must reflect expected internal supply of the sub block. In Fig 11, the violation can be resolved by relating the backup supply of the isolation instance to the sub-design interface port supply. If so, the path, from sub-design interface to top logic, can imitate full chip supply path. Then, the violation will be cleared.

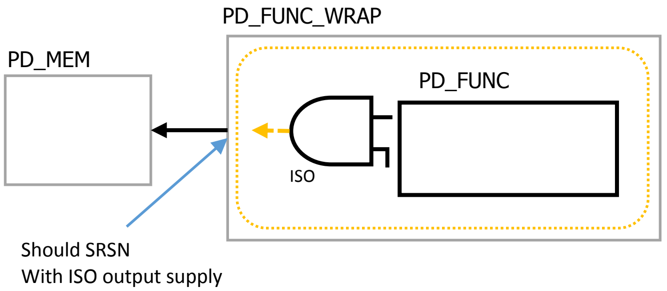


Figure. 11. ISO_INSTANCE_MISSING due to discarded sub-design logic. SRSN should be applied to the interface port.

Another case of ISO_INST_MISSING is flagged for unimplemented nodes. Full chip check detected all unimplemented nodes while top only flow missed some nodes depending on supply constraints at the interface ports. A block level implementation can be verified either by the block level verification or full chip verification. Top only flow does not cover what is supposed to be implemented at the sub-design level. It can manually model the impact of the protection instances such as isolation cells and level shifters by adding supply constraints SRSN/SPA. However, top only flow is not able to detect missing instances inside a sub-design as shown in Fig 12.

If the top verification engineer knew that a path to PD_WRAP_PLL_FUNC is not protected, then the interface port for the path could be related to a more ON supply of PD_WRAP_PLL_FUNC. Then the sub-design interface aware top only flow can flag the same violations as the full chip flow.

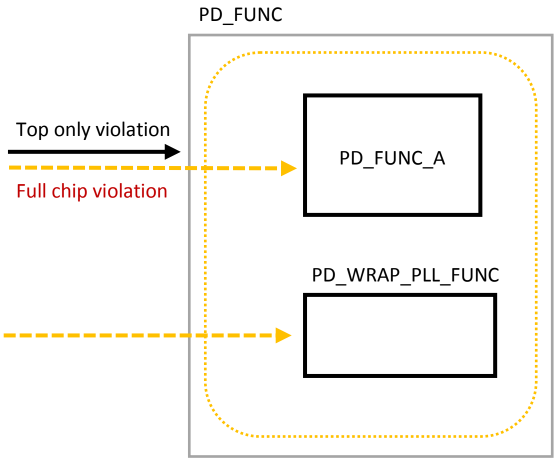


Figure. 12. ISO_INSTANCE_MISSING due to unimplemented isolation cell. The violation can be fixed at the block level.

Most SINK_SUPPLY_LEAKAGE occurred at the top logic, and are the same between the full chip and the top only flow. Additional SINK_SUPPLY_LEAKAGE flagged on the full chip side where the internal part of a sub-design contains bad supply connection.

LS_INST_MISSING produced the same check results between both flows for the path of a sub-design interface to top logic. Full chip check detected 22 more violation paths where violation paths reside inside sub-designs.

LS_INPWR_CONN and LS_OUTPWR_CONN are flagged when a level shifter supply mismatches the driver/load supply. These violations are caused by wrong supply constraints on the boundary ports. They can be easily fixed by putting the correct SRSN/SPA in the UPF.

Sub-design interface aware top only flow successfully detected (ISO_STRATEGY_UNUSED) unimplemented nodes from sub-design boundaries to top. Fig. 13 represents the violations points.

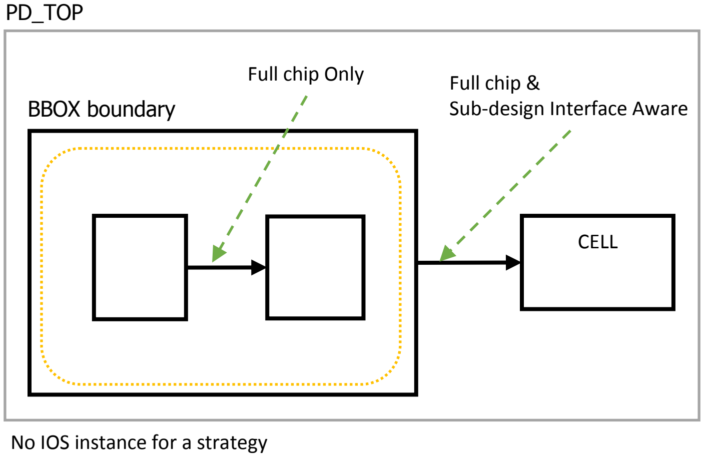


Figure. 13. ISO_STRATEGY_UNUSED identically detected for sub-design interface to top logic.

ISO_*_UNCONN and ISO_*_CONSTANT point out that isolation instances are either unconnected or driven by a constant TIE cell. These tags are properly reported at the top level. Full chip flow flagged additional violations for internal parts of sub-designs.

Unfortunately, LS_INST_REDUND violations stretching to the sub-designs are not able to be covered by sub-design interface aware top only flow. Full chip check detected two level shifters in a row, one in the block design and another at the top as shown in Fig 14. Even if the sub-design port SRSN/SPA is correctly assigned, top only flow has no information about whether a level shifter instance exists inside or not.

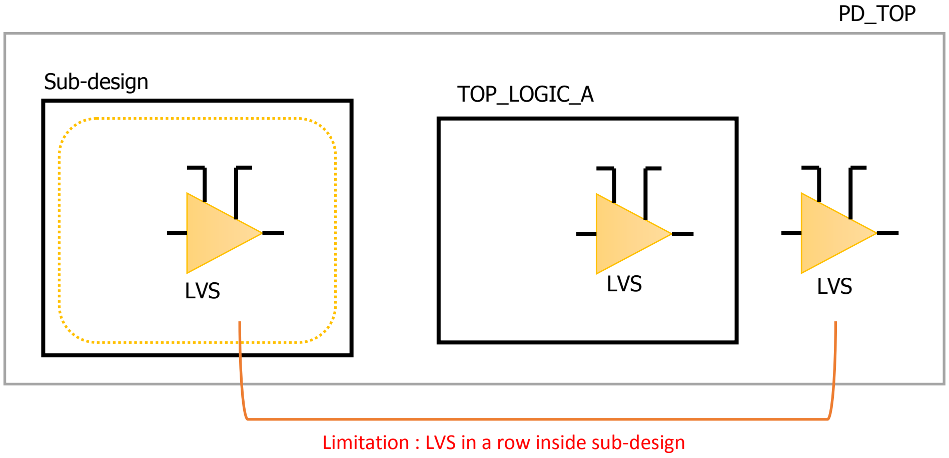


Figure. 14. LS_INST_REDUND that stretch to a sub-design.

LS_OUTPUT_UNCONN at the top level can be detected by both methodologies. The violation in the sub-design should be verified by block level verification or by full chip check.

TABLE 5
ROOT CAUSE COUNT OF INCONSISTENT RESULTS AT DESIGN CHECKS.

Violation Tag	Unique cases	Description
ISO_INST_MISSING	2	Isolation instance is missing from source to sink crossover
SINK_SUPPLY_LEAKAGE	1	Sink logic is ON where driver is OFF
LS_INST_MISSING	22	Level shifter instance is missing from source to sink crossover
LS_INPWR_CONN	1	Input power of level shifter is not compatible to source power
LS_OUTPWR_CONN	1	Output power of level shifter is not compatible to sink power
ISO_STRATEGY_UNUSED	1	No isolation cell present for an isolation strategy
ISO_OUTPUT_UNCONN	1	Isolation instance output is unconnected
ISO_DATA_CONSTANT	1	Isolation instance is driven by constant TIE
ISO_DATA_UNCONN	1	Isolation data pin is unconnected
LS_INST_REDUND	1	Level shifter instance not required
LS_OUTPUT_UNCONN	1	Level shifter instance has no load.

IV. CONCLUSION

We have demonstrated a sub-design interface aware top only static low power verification with the application processor to address two main problems in existing hierarchical static low power methodologies. First, full chip check that brings up top level violations can be performed only after sub-design verifications are done. Secondly, existing early stage top only verification flow requires manual black boxing and separate UPF sets that make it harder to control UPF flow for the project. In the presented new flow, properly assigned supply constraints, with SRSN/SPA, at sub-design interface ports are sustained automatically by VC-LP. Power intents around boundary ports and user intended supply constraints, which reflect the internal supply connections, guarantee the top level violation detection that is comparable to full chip check with one limitation. The limitation is that the flow is not able to detect instances inside a sub-design. Thus, LS_*_REDUND are not coverable for the path of a sub-design to top. By putting a forecast supply on the sub-design boundary, this methodology could match most of the top level violations to those of full chip flow.

Imitation of full chip path supply relationship by SRSN/SPA constraints determines the quality of the sub-design boundary to top logic check. This methodology enables parallel static low power verification between top and blocks. Further development in constrained attributes of sub-design interfaces, will overcome the limitation of this methodology.

ACKNOWLEDGMENT

Many thanks to the design team engineers involved in this collaboration work and Synopsys VC-LP R&D, CAE and Field engineers. The problem of hierarchical verification is addressed by mobile AP project. Synopsys had a great opportunity to apply a new flow to the AP design. Thank you for reading our methodology and hope to see you perform easier top level static verification.

REFERENCES

[1] H. Bhatt, Prashanth M, “Combining Static and Dynamic Low Power Verification for the Power-Aware SoC Sign-off”, DVCON-EUROPE 2014.

[2] D. Majhi, L. Shaotao, "UPF-Based Static Low-Power Verification in Complex Power Structure SoC Design using VCLP", SNUG Silicon Valley 2015.

[3] "VC-LP User Guide" https://solvnet.synopsys.com

[4] "VC-LP Message Reference Guide" https://solvnet.synopsys.com.