Left shift catching of critical low power bugs with Formal Verification

Manish Kumar, Madan Kumar, Madhusudhana Lebaka, Srobona Mitra
Qualcomm India Private Limited

Abstract- Low power verification is one of the last items to be closed in verification cycle. With increasing complexity of power-aware designs (100+ collapsible domains), it is challenging to validate all the power modes within the timeline constraints. Validating power-aware designs in simulation requires development of additional power sequence tests in testbench which is time and resource intensive. Also, the power-aware simulations at subsystem and SoC levels take days to complete, stressing the debug timelines. We have seen late bugs and ECOs related to low power features in multiple chips.

The power intent in UPF is instrumented by front end tools in RTL phase. It is important to validate low power features at this stage itself and catch as many bugs as possible, to save the cost of fixing late bugs. Therefore, we need complementary solutions to simulation to address the afore-mentioned problems. This has motivated to explore domains like formal verification to left shift and improve quality of power-aware verification.

In this paper, we present a methodology of leveraging formal methods for low power verification. With the help of the low power compile and elaborate capabilities developed in Formal tools, we have built applications to automatically generate and prove assertions for different low power specific problem scenarios. We specifically discuss 1) Power Aware Connectivity – to validate critical low power paths. 2)Automated low power checks using information model – to validate standard checks across all designs.

We see significant left shift and effort reduction compared to simulation with our formal methodology. This methodology can potentially help us to move towards full formal sign-off for low power features at block/IP level and to reduce simulation dependency in SoC/subsystem level verification.

I. CHALLENGES IN LOW POWER VERIFICATION

The power-aware SoC designs are increasing in complexity every year. Current chips have 100+ collapsible domains with different operational modes for each domain. With this level of power complexity, closing verification signoff before ECO stage is a major challenge.

Traditional power-aware simulation verification has requisites - development of low power testbench with required power sequences. Because of the required time and resources, low power verification is often skipped at IP level. Because of late enablement primarily at subsystem levels, we are left with limited time to catch bugs before it leads to expensive ECOs. Large runtimes for low power tests do not help with faster turnaround and debug cycles.

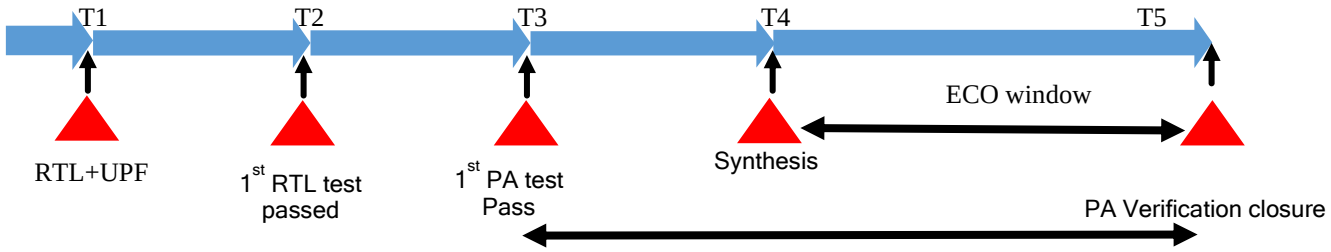


Fig 1. Timeline constraints for low power bug fixes

As we see in illustration Fig 1, actual PA functional simulation starts once RTL simulation and sanity power testcases are up and running. Often because of short window, bugs get slipped past RTL verification. It is important to left shift this verification process, so that any bugs in the design is caught ideally before we go to ECO stage.

II. FORMAL FOR LOW POWER

Formal is a great tool to address the afore-mentioned challenges with late verification closure. But traditional Apps in Formal tool did not support low power support. i.e., Formal could only be used to validate on non-power-aware designs. Connectivity and standard RTL checks without UPFs for power critical paths will not be accurate and have to depend on functional simulation for power related issues.

In this paper, we present novel formal low power methodology and corresponding vendor collaborations to get tool ready to support power features and

As part of this initiative, we have worked with tool vendors to support compile and elaboration of UPF+RTL in the tool. Since formal would only require the design (RTL+UPF) without any testcase requirements, we could enable formal much earlier than simulation. This allows for increased quality UPF release for simulation stage. Also, few of the problem scenarios can be signed off with formal alone. This will significantly left shift and avoid ECOs. Through connectivity checks at start of integration and low power checks at IP level, few weeks of debug effort is reduced at SoC level.

Below figure illustrates the projected left shift with the deployment of this flow. Formal flows can be intercepted as soon as UPF is available.

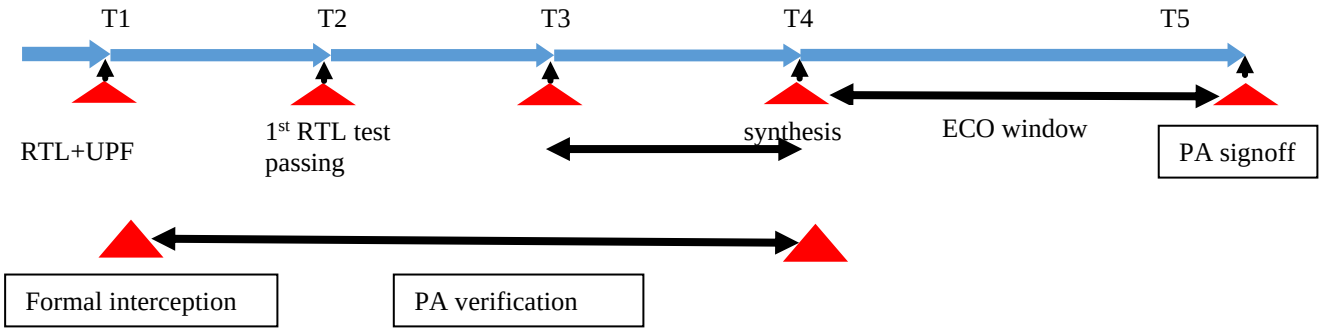


Fig 2. Early interception with Formal

III. FORMAL LOW POWER FLOW

We have collaborated with tool vendors to support power-aware compilation. The RTL+UPF is compiled and elaborated to get a low power database. We have further worked with vendors to get connectivity use model and information model support for our applications. With this as base we have built low power flows to automatically generate assertions for different problem scenarios. User has to provide specification/ configuration in a required format. This will then be converted to properties and validated in tool through formal proof. The failing properties are then debugged to see if any low power issues and fix them.

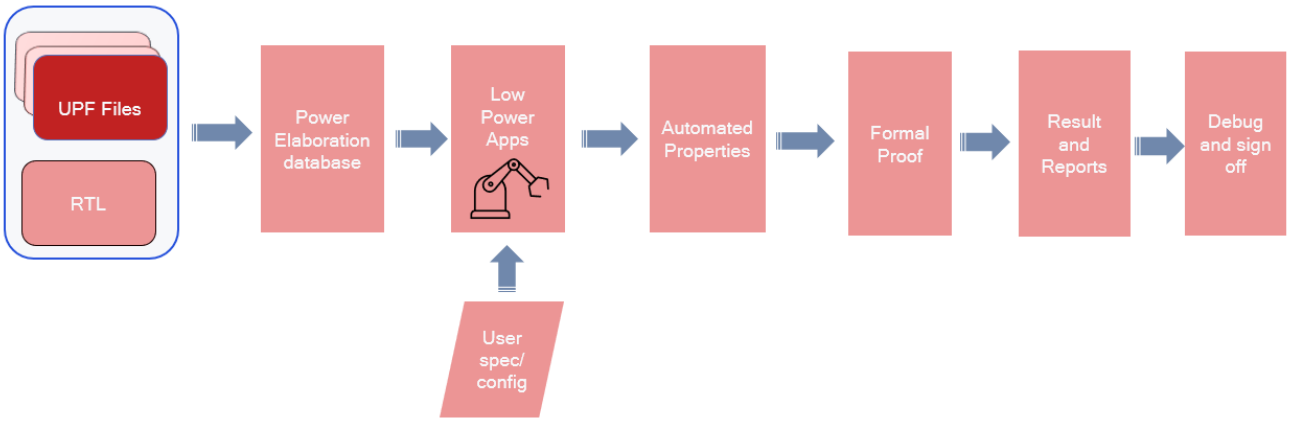


Fig 3. Architecture of formal low power methodology

IV. Applications

With general framework of low power flow, we have built applications that validate different low power scenarios. We present two applications current implemented in Qualcomm

1. Power Aware connectivity
   We will have multiple power rails at SoC level. For instances such as memories or standard cells, specific rails are required to be connected from top. With power aware connectivity, this kind of expectations can be validated. Also, any controlled paths such as power switch connectivity can be validated. The other use case in RTL connectivity with specific power element behavior in the path. There are standard paths where certain isolation behavior is expected. These kinds of checks can be easily validated with this flow. The expected user spec is in standard connectivity formats like CSV. This allows for reuse and compatibility with existing non-power connectivity flows.

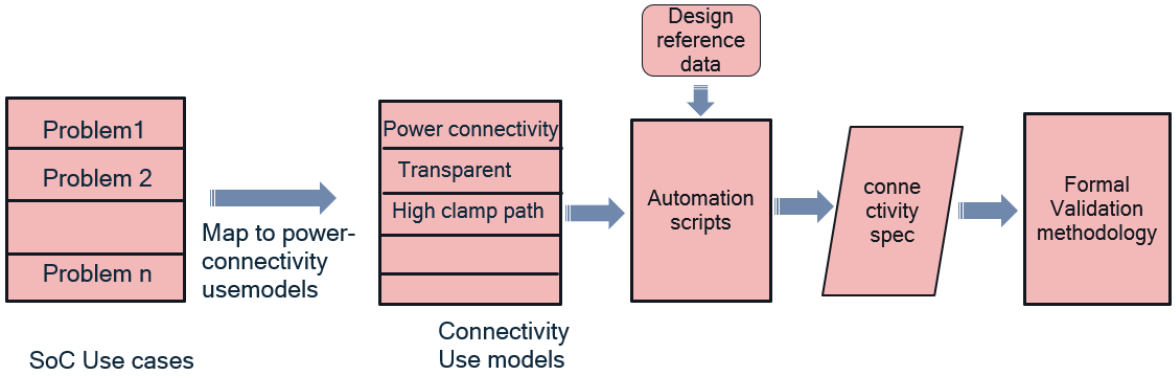


Fig 4. Power aware connectivity flow

2. Automated standard low power checks

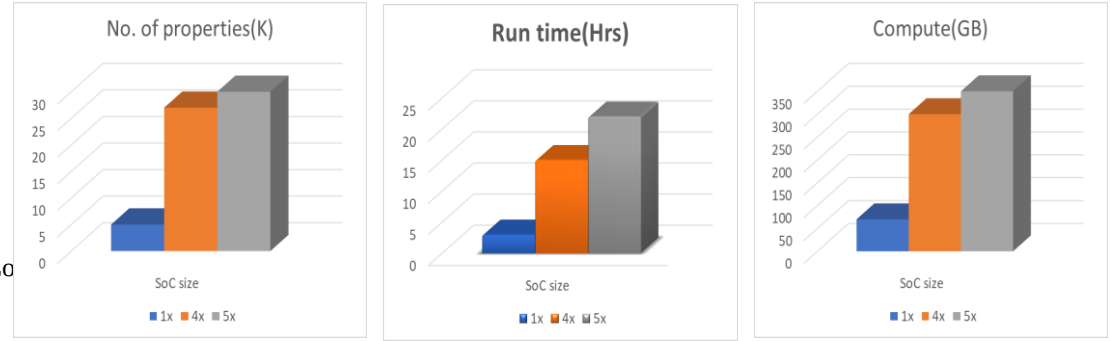
The low power policies in UPF cannot be directly accessed in testbench to monitor or bind and check assertions. We have information model standards with query commands and bind APIs defined in UPF LRM[1]. Once the power elaboration support was available in the tool, we have worked with vendors to define the query commands and other API support needed for our applications. We have then built an application to leverage information model APIs to build standard SVA assertions. The assertions can be validated across designs for stand low power elements. In simulation, we need development of full pledged low power testbench to validate these checks. With formal, we are able to validate the checks with minimal constraints to drive power sequence. The flow is mainly implemented at IP level and the results are reusable at higher integration verification using simulation. This saves effort in debug and fixing bugs at later stage,

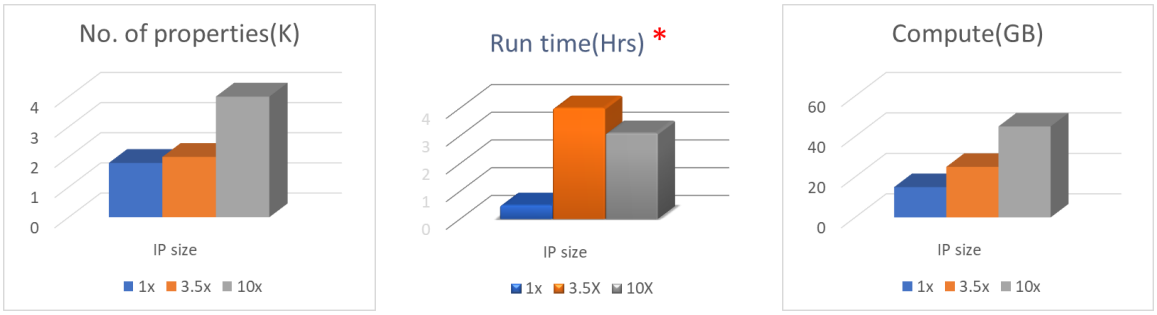
V. Validation Results

1. Application of connectivity app

This flow is deployed in multiple chips at SoC level. We are able to validate 1000s of connections in a small time. The result metrics are plotted below –

2. Lo





This flow is validated at IP and smaller subsystem level. The COI and power complexity is relatively bigger than connectivity checks.

VI. Conclusion

Exhaustive low power connectivity checks are run on SoC designs. This saves debug cycles that were used in fixing the UPF issues during SoC integration. Verification of information model-based checks left shifts the verification by catching bugs at smaller IP/HM level. Bring-up time reduction to 60% seen compared to simulation for same checks. Left shift of about 2-3 weeks seen with our current bring-up examples. We are working towards more formal adaption in power-aware verification through development of newer applications and further automations. We see potential for exclusively signing off few IPs with Formal and to reduce simulation dependence at subsystem and SoC integration levels.

Acknowledgment

Multiple teams within Qualcomm (project DV, methodology) have worked in the development/validation of the flows. Thanks to tool vendors in collaborating to develop the tool infrastructure required from scratch up.

References

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