2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Reducing Area and Leakage Power: Novel Formal Methodology for Retention Sufficiency in Low Power Designs

Madan Kumar, Chepuri Venkatesh, Durga Prasad, Kuber Derasari, Srobona Mitra, Nitin Neralkar
Qualcomm India Pvt Ltd, India (mkumarr@qti.qualcomm.com)

Jimik Shah
Qualcomm Technologies, Inc

Abstract—Retention cells are essential in low-power designs to preserve logic states during power domain shutdowns, but they introduce significant area and leakage power overheads. Traditional simulation-based verification methods often miss corner cases, leading to over-retention and degraded PPA. This paper presents a novel formal verification methodology that combines power-aware formal tools with Sequential Equivalence techniques to validate retention sufficiency. The proposed flow includes design constraints, formal VIPs, automated debug loops, and convergence strategies to scale verification for large IPs. Experimental results demonstrate 30–50% retention savings with reduced runtime and verification effort, enabling early and efficient validation of low-power architectures.

I. Introduction

Low Power designs are extensively used in various applications – mobile and consumer electronics, Automotive, compute, IoT and edge devices. For battery operated devices such as mobile chips, the design complexity is very high. Low-Power architecture is used for these chips to reduce power consumption. The design is divided into independently operated power domains that can be turned OFF based on the operation. To implement these power domains, we need additional low power elements such as isolation, retention, level shifters, power switches etc.

State Retention is a key low power technique to reduce latency impact, when a power domain resumes normal operation. The logic state of the design is saved through retention cells before Power OFF and loaded once Powered ON. This will save clock cycles required to re-initialize the design and ensure seamless operation.

II. Retention complexity and verification challenges

Design complexity

Retention implementation requires millions of retention cells in place of normal flops. This also requires additional circuitry in the form of buffers, backup supply network, and connectivity for control signals. This adds overhead in terms of area and routing congestion/complexity. Another major Overhead is in terms of leakage power. Retention cells will be active always and have constant leakage current even when the domain is switched OFF. This adds up the significant power leakage in the chip. With increasing chip size and complexity, these overheads are becoming major constraints and hence need to be minimized.

Traditionally simulation is used for retention verification. It is difficult to analyze and implement all the retention use-cases. Often corner case and software scenarios are missed in cores when partial retention is

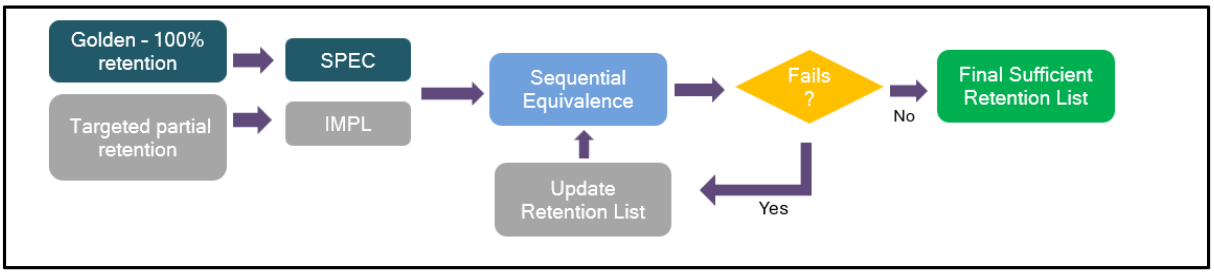
implemented. This leads to ECOs that can restrict low power features or lead to additional verification effort. Architects often lack the tools needed to correctly analyze retention requirements and tend to over-retain. This degrades the chip PPA in terms of area and power.

III. Formal solution

Formal verification addresses several challenges seen in simulation. Formal verification involves exhaustive verification for specific problem scenarios avoiding any bug misses. Also, the TB effort for formal is minimal as there is no need to implement functional scenarios/stimulus manually.

We do not have a formal application that can be used to verify retention sufficiency. Verifying retention sufficiency requires us to prove the equivalency of implemented partial retention policy with full retention for the power domain. While Sequential Equivalence is a known Formal technique (used in applications such as clock gating verification), it doesn't have the capability to verify low power features of the design. In this paper, we present novel power-aware formal flow for retention verification. Here we combine the capabilities of Low Power Verification Formal app with Sequential Equivalence. We have developed retention specific optimizations in the flow.

The low power capabilities required is developed in the tool through our (Qualcomm) collaboration with EDA formal tool vendors. The following flow is implemented in power-aware Sequential Equivalence flow



Retention Sufficiency Verification flow

The power domain IP with full retention is compiled as SPEC. In the targeted IMPL, partial retention version of same design is compiled. The Sequential Equivalence checks are set up at the time of retention restore. The checks are implemented at the outputs of design-domain boundary. The non-retention flops in the IMPL design will have reset values as part of power sequence. If the non-retention flops impact the normal functional behavior of the IP, the corresponding output checks will fail. These failing counterexamples (CEX) are debugged to identify the missing/incomplete retention.

Flow Optimizations

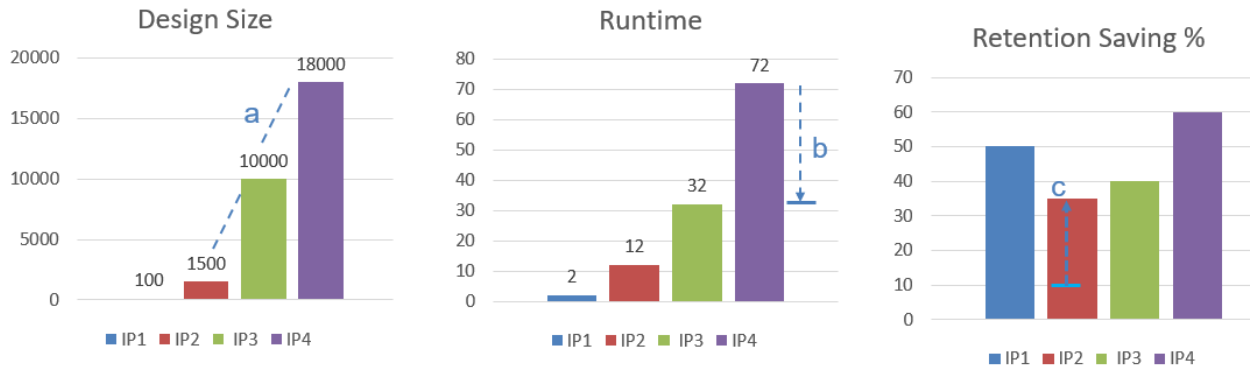
The vanilla flow described above has several limitations. The retention saving obtained is very minimal (5%) due to exhaustive verification. Also, the flow can become iterative as the initial retention list is limited by designer knowledge of low-level flops and can have multiple missing retention. Another major challenge is in terms of design size complexity. Because of additional low power complexity in the Sequential Equivalence setup, even designs with few thousand flops will have convergence issues.

Below Optimizations are implemented for the flow to scale to bigger designs and with reduced verification effort.

1. Design constraint flow – Power domain IPs often should satisfy conditions for it to be eligible for power collapse. It can be in terms of a known idle state or operations like clearing pipelines and buses. These conditions are integrated in the flow to eliminate redundant retention for invalid functional scenarios.
2. Formal VIP is implemented to drive the standard power sequences required for validating checks. This is a plugin to avoid TB effort to implement formal stimulus.
3. The debug of fails and the iteration loop is automated so that user intervention is not required during the runs.
4. Standard convergence techniques to intelligently divide complex designs into manageable setups.
5. End-to-End flow with all the automations and user-friendly reporting

IV. RESULTS

The flow was validated in multiple designs. Based on the results obtained, we have optimized the flow for better ROI in terms of performance and optimized results. Below graphs show the KPIs for runtime, design sizes and the retention savings.



a. To scale for designs beyond 10K, we implemented multiple convergence and divide-and-conquer methods in the flow.
b. For IP size of 18K flops, we see the runtime shoots up beyond 72 hours. With proof optimizations developed, we were able to reduce runtime by more than half. We see a slight tradeoff in retention savings (5%) with these optimizations. We are working on flow changes to reduce these deltas.
c. We see consistent saving of 30-50% retention saving in the validated IPs compared to legacy full retention. The results show great promises for wider deployment and significant retention saving in the chip.

V. CASE STUDY - RETENTION SUFFICIENCY FLOW FOR POST-SILICON ROOT CAUSE ANALYSIS

In one SoC, a post-silicon issue was found where the Finite State Machine (FSM) of the Design Under Test (DUT) incorrectly transitioned from SLEEP to WAKE-UP and then back to SLEEP state. This rapid

switching between power states caused system instability and higher power consumption. Designers suspected partial retention in the DUT as the cause.

To root cause this, deployed the formal Power Analysis Sequential Equivalence Checking (PA SEC) flow where the specification DUT was fully retained, while the implementation DUT has silicon retention elements. We elaborated at DUT level, and did the PA equivalence for the FSM state. We observed a glitch in the clock gate (clk_gate) and power gate (power_gate) logic which are in COI of the FSM. The clk_gate and power_gate are connected to the DUT FSM through a synchronizer. The synchronizer inputs are retained but the synchronizer flops are not retained. As a result, when the power-up sequence started, the synchronizer inputs, clk_gate and power_gate are asserted, the synchronizer outputs initially held a reset value (1'b0). It took 2-3 cycles for the actual retained input values to propagate to the FSM. This delay in propagation led to the temporary and erroneous transition of the FSM state from SLEEP to the WAKE-UP state. This issue is fixed and validated using the PA SEC flow

VI. Conclusion

A novel low-power Formal flow is implemented to verify retention sufficiency of low power designs. This flow addresses the limitations of simulation for IP level designs and left shifts retention verification by 3 weeks. Retention saving of 30-50% observed for fully retained IPs. Controlled deployment for few chosen blocks is in progress, with plans to scale to bigger and wider designs to target area and power savings in the chip.

Acknowledgment

We thank multiple DV teams within Qualcomm that have helped in validating the flow and sharing feedback for optimization. We acknowledge the support from EDA vendors for their collaboration in developing the flow.

References

[1] "IEEE Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems," in *IEEE Std 1801-2018* , vol., no., pp.1-548, 29 March 2019, doi: 10.1109/IEEESTD.2019.8686430