

Achieving Glitch Power Visibility via Delay-Shifted Analysis for Actionable Power Optimization

Neetu Mehta, Chris Moench, Kathy Hoover, Sandeep Vundavalli

AMD India Pvt Ltd, India

Corresponding Author: Neetu.Mehta@amd.com

Abstract: Glitch power, the dynamic power consumed by spurious signal transitions before outputs settle, is a significant contributor to total power in modern SoC designs, yet it remains invisible to conventional RTL power analysis. Traditionally, capturing glitch power requires SDF annotated gate-level simulation, imposing prohibitive runtimes and repeated design verification handoffs. This paper presents a delay shifted waveform analysis methodology which enables glitch power estimation, including glitch generation, propagation, and configurable filtering from zero delay waveforms without SDF annotation. This work evaluates the application of delay-shifted event analysis, an EDA power analysis technique available in production signoff tool, to quantify and address the glitch power blind spot in conventional analysis flows. The approach dynamically adjusts waveform edges based on calculated arrival times, revealing glitch behavior that is otherwise hidden. Evaluation on an advanced tech node demonstrates that delay shifted analysis captures glitch power with a fidelity approaching SDF gate-sim while reducing turnaround from weeks to hours, enabling faster iterations during power optimization flows.

Keywords: *Glitch Power, Delay Shifting, Power Analysis, Without SDF.*

I. INTRODUCTION

Power analysis is a cornerstone of the SoC design flow at advanced process nodes. As transistor densities increase and operating frequencies rise, the accuracy of power estimation directly impacts critical design decisions: thermal management strategies, power delivery network sizing, and the trade-off between performance and power efficiency. An underestimate of power leads to thermal violations or silicon failures; an overestimate leads to excessive design margins and lost performance.

At advanced nodes, the glitch power problem intensifies. Increased logic depth in combinational paths, tighter timing margins, and the growing disparity between gate delays and interconnect delays create more opportunities for signals to arrive at different times, generating more frequent and wider glitches. Industry analysts estimate that 20-40% of total dynamic power can be attributed to glitch activity in advanced designs, making glitch-aware analysis essential rather than optional.

Among the components of dynamic power, glitch power is particularly challenging to estimate. A glitch is an extra, unintended signal transition that occurs before a gate output settles to its final logic value within a clock cycle. Glitches

arise because different input paths through combinational logic have different propagation delays. When a faster path causes a gate output to transition momentarily before a slower path arrives and establishes the correct output, the result is a spurious toggle, a glitch that consumes dynamic power. Moreover, glitches can propagate through downstream logic, creating cascading transitions that amplify the power impact.

The conventional approach to capturing glitch power requires SDF (Standard Delay Format) annotated gate-level simulation. SDF is an IEEE standard that specifies cell delay, interconnect delay, and timing checks. By annotating the gate-level netlist with these real delays (generated via the `write_sdf` command from Primetime, with a separate file per timing corner), the simulation models actual signal propagation timing, revealing where glitches occur. However, this approach imposes severe practical limitations: the gate-sim loop alone often takes days, while the full SDF version can extend to weeks. By the time results are available, the design has typically evolved, rendering the power analysis stale.

This paper presents a delay-shifted waveform analysis methodology that enables glitch-aware power estimation without SDF annotation, reducing multiple steps for iteration while maintaining accuracy approaching SDF gate-sim. The methodology is implemented in Power analysis and integrated into the flow which will help to do further analysis.

II. PROBLEM STATEMENT

A. The Glitch Power Blind Spot

The power analysis life cycle progresses through four stages of increasing accuracy: (1) RTL average-mode, (2) RTL time-based, (3) gate-level simulation, and (4) SDF gate-level simulation. As accuracy increases at each stage, runtime also increases significantly, limiting the ability to provide meaningful feedback for RTL or physical design changes.

RTL average-mode analysis estimates power using two statistical values annotated on primary ports and register outputs: static probability (SP), the probability that a signal is at a specific logic state (expressed between 0 and 1), and toggle rate (TR), the number of logic transitions per unit time. These values are propagated through downstream combinational logic cones using proprietary statistical analysis to determine which conditions trigger multi-input gates. Default toggle-rates can be used where signal annotation is missing. However, this statistical propagation fundamentally cannot model glitch behavior because it lacks timing information, the tool knows how often signals toggle, but not when they toggle relative to other signals arriving at the same gate.

RTL time-based (cycle-accurate) analysis improves on this by propagating actual waveforms from primary inputs and registers to calculate power conditions per cycle. This analysis requires full annotation and captures real switching patterns. However, because it operates on zero-delay waveforms where all signals transition simultaneously at clock edges, it still cannot reveal the timing-dependent glitch behavior that occurs when signals arrive at different times due to path delay differences.

B. Effects

Traditionally, many design teams rely on stages 1 and 2 of the power analysis lifecycles, where glitch power is not explicitly characterized. This trade off enables simpler, faster analysis flows while limiting visibility into glitch-related power effects. A few design teams extend power analysis to stage 3 (gate-level simulation without SDF) to improve accuracy while reaching stage 4 (SDF gate-sim) for glitch power analysis remains impractical for all teams due to multiple steps in that stage and the limitation of being able to run only a few iterations. The gate-sim loop takes days; the SDF version can take weeks. The result is that glitch power which can be a meaningful contributor to total dynamic power is effectively invisible to design teams during the active design cycle when optimizations can still be made. While glitch power analysis can provide guidance on high-power areas and suggest logic restructuring in RTL or physical design, the information arrives too late to be actionable.

III. PROPOSED SOLUTION

A. Delay Shifting: Core Concept

Delay Shifting is a feature which can be used to enable glitch-aware power analysis from zero-delay waveforms. The core idea is straightforward: instead of requiring an SDF-annotated gate simulation to model real signal timing, Prime Power takes a zero-delay input waveform from either RTL simulation or zero-delay gate-sim and dynamically adjusts the waveform edges based on the calculated arrival times derived from the gate-level netlist timing information already available in the Prime Power session.

This delay-shifted waveform reflects realistic signal transition timing. Where the original zero-delay waveform shows all gate outputs changing simultaneously at clock edges, the delay-shifted waveform shows outputs transitioning at different times based on their actual path delays. This timing separation is precisely what creates glitch conditions, when a fast path causes a gate to produce an initial output that is then overridden when a slower path arrives. By reconstructing these timing relationships, delay shifting reveals glitch behavior that is completely hidden in zero-delay analysis.

B. How Delay Shifting Works

The delay shifting process operates as follows. First, Prime Power loads the gate-level netlist along with its timing information (liberty files, parasitic data). Second, the zero delay input waveform is loaded with the functionally correct switching activity but without timing information. Third, it calculates the arrival time at each gate output based on the cell delays and interconnect delays in the timing model. Fourth, the tool shifts each signal transition in the waveform to its calculated arrival time, producing a delay-shifted waveform that approximates what an SDF gate-sim would produce.

Finally, power analysis is performed on this delay-shifted waveform, capturing both the intended switching activity and the glitch transitions that emerge from the timing relationships.

Conventional approaches typically involve multiple sequential steps, including RTL simulation, gate level simulation, timing annotation generation, SDF annotated gate level re-simulation, and subsequent power analysis. This multistage process requires explicit SDF files for each timing corner and often results in limiting iteration throughput.

In contrast, the delay-shifted flow streamlines the analysis by operating directly on RTL or zero-delay gate-level simulation waveforms and incorporating delay shifting within the power analysis stage to produce glitch-aware results. This reduced sequence of steps eliminates the need for SDF generation and annotation, simplifying setup and reducing dependency on corner-specific timing data. As a result, the delay-shifted methodology enables supporting multiple analysis runs within a day and thereby helping to improve design productivity.

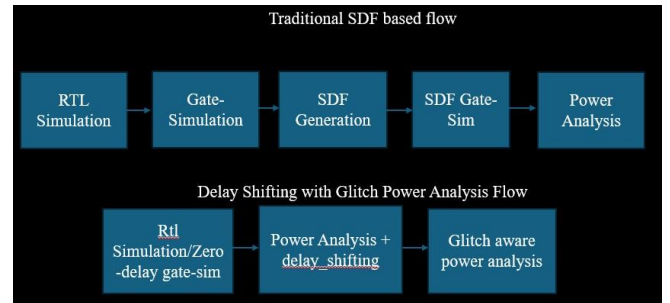


Figure 1: Flow comparison. Top: Traditional SDF-based flow with repeated design-verification handoffs. Bottom: Delay shifting flow completing with no SDF dependency.

C. Glitch Generation and Propagation

Once signal transitions are shifted to their calculated arrival times, glitch conditions naturally emerge at gates where inputs arrive at different times. Consider a two-input AND gate where input A arrives 50ps before input B. If A transitions from 0 to 1 while B is still at 1 (from the previous cycle), the output momentarily goes high before B transitions to 0 and pulls the output back down. This momentary high pulse is a glitch an extra toggle that consumes dynamic power.

Critically, glitches generated at one gate can propagate through downstream logic. If the glitch at a gate output is wide enough relative to the downstream gate delay, it will cause the downstream gate to also produce a glitch, creating a cascading effect that can amplify the power impact through the logic cone. This propagation behavior is modeled in the delay-shifted analysis.

D. Glitch Filtering via glitch_pulse_r

Not all glitches at a gate output will propagate to downstream logic. A narrow glitch may be absorbed by the inherent delay of the downstream gate the output of the downstream gate simply does not have time to respond to such a brief input disturbance. This physical filtering effect is modeled in power analysis through the glitch_pulse_r variable.

The `glitch_pulse_r` parameter specifies the filter ratio of glitch pulse width to gate delay, with valid values between 0 and 1.0 (default 1.0). The filtering mechanism is defined by the following propagation conditions, where a glitch at a gate output will propagate through a downstream gate only when:

$$w_glitch \geq d_gate \times glitch_pulse_r \dots\dots\dots (1)$$

Where `w_glitch` is the width of the glitch pulse in time units, `d_gate` is the propagation delay of the downstream gate from the timing model.

A glitch at a gate output propagates to the downstream gate only when the condition in equation (1) is satisfied. For example, if `glitch_pulse_r` = 0.5 and the downstream gate has a delay of 10ps, only glitches wider than 5ps will propagate through that gate. Glitches narrower than 5ps are filtered out, modeling the physical behavior where the gate cannot respond to such brief transients.

The default value of 1.0 means only glitches wider than the full gate delay propagates a conservative setting. Setting `glitch_pulse_r` to 0.0 disables filtering entirely, allowing all glitches to propagate. The optimal value depends on technology and library characteristics.

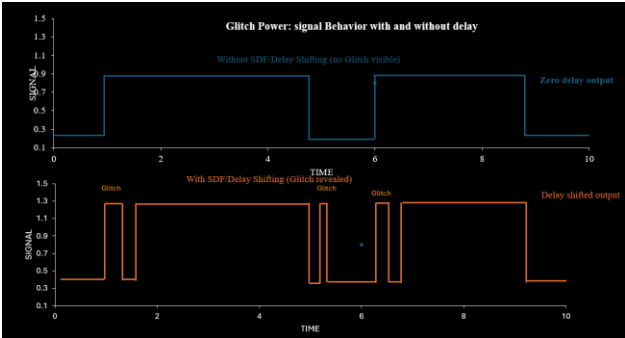


Figure 2: Signal waveform comparison. Top: Zero-delay waveform showing only intended transitions (2 toggles). Bottom: Delay-shifted waveform revealing glitch transitions (6 toggles including 3 glitches) that contribute to additional dynamic power consumption.

E. Enablement of delay_shifting Methodology:

A key advantage of the delay-shifted approach is its minimal configuration overhead, which makes it well suited for adoption in existing design flows. Full glitch aware power analysis can be enabled through a simple activation that builds directly on the standard time-based power analysis setup. Rather than introducing a separate workflow, the delay shifting mechanism is incorporated as an additional analytical layer on top of the baseline flow.

This integration does not require any changes to the simulation environment, stimulus generation, or netlist representation, thereby preserving compatibility with established verification practices. As a result, designers can obtain enhanced visibility into glitch-induced power effects with minimal disruption, enabling more accurate power characterization without increasing setup complexity. Collectively, these features operate as part of an integrated power analysis framework, as summarized in Table 1, without requiring modifications to simulation inputs or netlist configuration.

Table1: Below are the configurations it activates.

List of Options Activated	
power_enable_separate_glitch_report = true	power_enable_delay_shifted_event_analysis = true
power_enable_clock_cycle_based_glitch = true	power_delay_shifted_include_unit_delay = true

Further, Glitch filtering is controlled via another one variable in power and using that variable there is a primary option is `glitch_pulse_r` followed by a numeric value. Additional lesser-used options are available for fine tuning specific analysis behaviors which help to give realistic glitch number.

IV. EXPERIMENTAL RESULTS

The delay-shifted glitch analysis methodology was evaluated on an advanced tech node design. The evaluation compared three analysis configurations: (1) standard RTL time-based analysis without delay shifting (baseline, no glitch visibility), (2) delay-shifted analysis with `glitch_pulse_r` = 1.0 (default, conservative filtering), and (3) delay-shifted analysis with `glitch_pulse_r` = 0.5.

The results demonstrated that delay-shifted analysis captures glitch power contributions with fidelity approaching the SDF gate-sim reference. Regions of the design identified as high glitch-power areas by the delay-shifted analysis correlated with those identified by the full SDF flow, validating the accuracy of the arrival-time-based approach. The separate glitch report provided per-instance and per-net glitch power breakdowns, enabling targeted identification of logic structures most susceptible to glitch induced power overhead. The iteration has increased by reducing the number of steps that enable designs to perform glitch-aware power analysis within the active RTL and physical design development cycle. This represents a shift from glitch power being a post-implementation verification metric to an actionable optimization metric available during active design.

Table 2: Glitch Filtering Sensitivity (Based on Advanced Tech node)

glitch_pulse_r	Glitches Detected	Glitch Power	Behavior
1.0 (default)	Fewer	~ 81 mW	Conservative - filters narrow glitches
0.5	More	~ 135 mW	Balanced - captures wider range of glitches

Table 3: Impact of Glitch Power vs Total Power

Configuration	Total Power	Glitch Power	Glitch % of Total
Time-Based (no delay shift)	~338 mW	0 (invisible)	0% (blind)
Delay Shift (R=1.0)	~ 419 mW	~ 81 mW	~19%
Delay Shift (R=0.5)	~ 472 mW	~ 135 mW	~28%

Where, R is glitch pulse rate in table.

V. RELEVANCE

Glitch power has long been a limited-visibility aspect of conventional power analysis flows. Usually designs, operating in stages 1-2 of the power analysis life cycle, have had no practical means of estimating glitch power contributions. In other case design also use stage 3 to capture some glitch effects through gate-level simulation but without the SDF timing that reveals the full picture. The delay-shifted methodology addresses this gap by making glitch-aware power analysis accessible to all without changing their existing simulation. The integration within the existing flow of the standard power analysis framework across product lines ensures that adoption requires minimal effort. Designers can use this feature for their time-based power run configuration. The resulting glitch power reports provide actionable data for identifying logic structures and physical design configurations that generate excessive glitch activity. While it is acknowledged that proving whether physical design optimizations to reduce glitch power directly impact silicon behavior remains challenging, the methodology provides directional guidance that is valuable for power critical designs. The practical benefit is immediate: design teams can now identify and address potential glitch power issues during the active design cycle rather than discovering them post-silicon or not at all.

For example, delay-shifted glitch reports on the advanced tech node design identified specific logic structures in the data paths where cascading glitches contributed disproportionate dynamic power. This visibility enabled the design team to evaluate targeted optimizations such as pipeline register insertion to break glitch propagation paths and buffer rebalancing to reduce arrival time skew at critical gates optimizations that were previously not considered because the glitch contribution was unknown.

VI. CONCLUSION

This paper addressed a critical gap in conventional power analysis methodologies by highlighting the limited visibility of glitch-induced power under standard time-based approaches. To overcome this limitation, a delay-shifted waveform analysis methodology was presented that enables accurate reconstruction of realistic signal timing while operating on zero-delay simulation waveforms. By leveraging timing information already available within the power analysis framework, the proposed approach avoids the need

for SDF-annotated gate-level simulations and their associated significant turnaround times.

Experimental evaluation on representative advanced-node designs demonstrated that the proposed methodology is effective in identifying regions of elevated glitch activity that are otherwise missed by conventional analysis. The results indicate that glitch-induced power can constitute a substantial portion of total dynamic power in modern designs, highlighting the importance of incorporating delay-dependent effects into power analysis.

An important aspect of the proposed approach is its practicality and ease of integration. The delay-shifted analysis is structured as an incremental enhancement layered on top of existing time-based power analysis flows. It does not require modifications to simulation environments, design representations, or verification infrastructure, thereby preserving compatibility with established design and analysis practices. This low overhead integration model enables broader applicability across different design styles and scales.

VII. REFERENCES

- [1] N. H. E. Weste and D. M. Harris, "CMOS VLSI Design: A Circuits and Systems Perspective," Addison-Wesley, 4th ed., 2011.
- [2] J. Rabaey and M. Pedram (Editors), Low Power Design Methodologies. Kluwer Academic Publishers, Boston, MA, 1996.
- [3] https://spdocs.synopsys.com/dow_retrieve/qsc-u/dg/ppolh/U-2022.12/ppolh/pdf/ppug.pdf
- [4] A. Bellaouar and M.I. Elmasry, Low-Power Digital VLSI Design: Circuits and Systems, Kluwer Academic Publishers, Boston, 1995.