

Decoding Glitch Power at the RTL Stage

A shift-left approach for glitch power estimation and optimization

Introduction

In the context of analyzing digital semiconductor circuits, a glitch is any unwanted or unused signal transition, or toggle. A glitch is often a transient signal that is much shorter than a clock period and therefore is not captured by the next register stage.

We also encounter full transition glitches, or transport glitches, which refer to toggles in a data path circuit that cover a full clock period but do not contribute to the overall functionality of data path logic. These incidental toggles are considered unwanted and are termed “glitches.” The complexity of most chip designs means they generate a large number of glitches, which, like all transitions, briefly draw current and consume power.

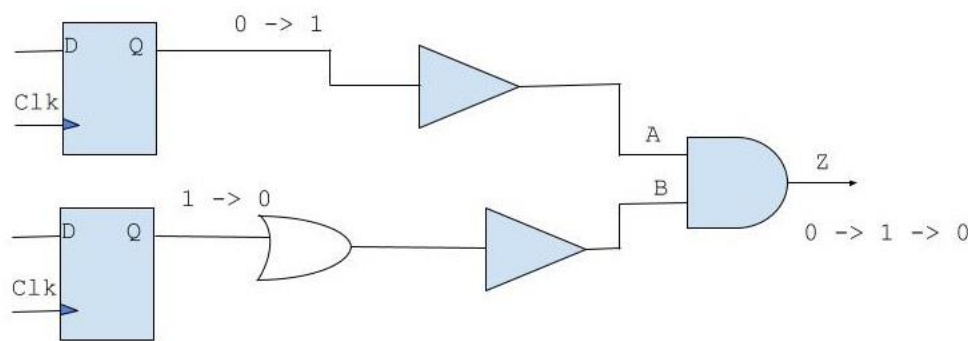
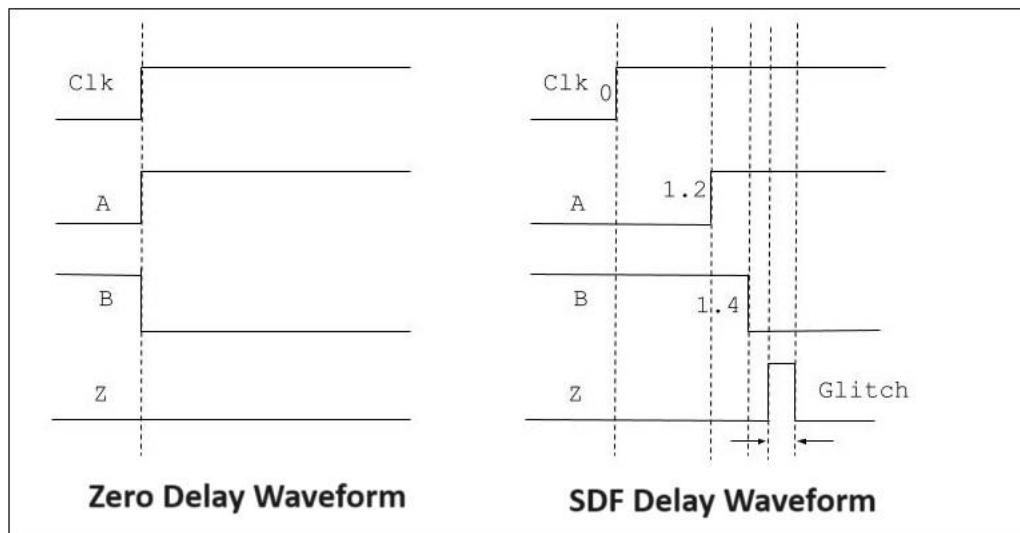


Figure 1. This example shows a common situation of a convergent data path where signals A and B combine to create output Y. In an ideal zero-delay analysis, Y should remain low. But with real delays, the different arrival times, or race condition, of A and B mean that Y briefly will rise high. This is a glitch.

If the incoming signals to a digital gate have different path delays, they can cause a glitch at the output, as shown in Figure 1. In designs with a lot of combinational logic, glitch power can represent 30% to 40% of the total power consumption of the circuit. Therefore, it is important to consider glitch power in the overall power consumption of digital circuits.

What Is Glitch Power?

When the circuit toggles, the state changes from logic 1 to logic 0 or vice versa. This change in state results from charging or discharging of capacitance nodes in the circuit. Each charging/discharging cycle consumes power, whether the toggle is desired or a glitch. Circuit power can be reduced by avoiding the glitches or minimizing their propagation from source to downstream logic stages.

Identifying glitch events is challenging, making it difficult to accurately pinpoint glitch power. There is no proven golden formula to compute glitch power; even gate-level glitch power does not precisely model the silicon effects. Yet, it is necessary to identify glitch-prone logic early in the design to minimize the impact of glitch power on the power efficiency of digital circuits.

Two Types of Glitches in Digital Circuits

There are two main types of glitches: partial transition glitches and full transition glitches. Both types can cause unnecessary power consumption in the circuit, leading to wasted glitch power.

Partial transition glitches

These are defined as consecutive transitions on a signal that are separated by a time interval smaller than a given threshold and where the transition is not a full signal transition—for example, a low-to-high transition that falls back before reaching high or a high-to-low transition that fails to reach zero volts before rebounding.

The time threshold usually is set to the average of the nominal signal transition times: $(\text{rise transition} + \text{fall transition})/2$. Another option is to set the threshold equal to the typical cell delay time.

The power consumed by a partial transition glitch is less than a normal transition because the load capacitances are only partially charged. The partial glitch power is calculated as the nominal transition power scaled down by a factor equal to $(\text{glitch pulse width}/\text{threshold time})^2$.

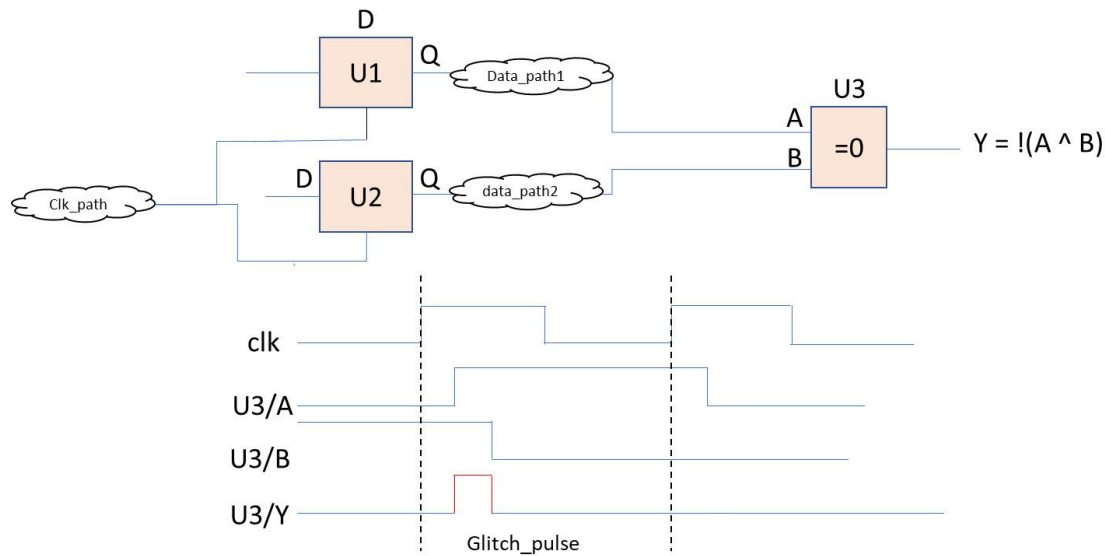


Figure 2. Example of a full transition glitch on signal u3/y

Full transition glitches

Full transition glitches are defined as consecutive transitions on a signal that occur more than once in a full clock cycle (threshold = one clock period). However, for clock signals, multiple transitions in just half a clock period count as a glitch (threshold = $\frac{1}{2}$ clock period).

No power scaling is done for full transition glitches. They consume the full, nominal dynamic power for each transition.

TG vs IG – Representation

- TG: More than 1 event in clock cycle
- IG: Pulse width < given threshold

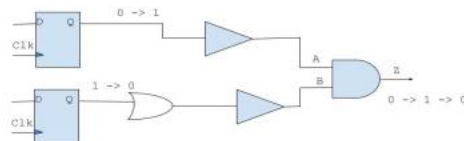
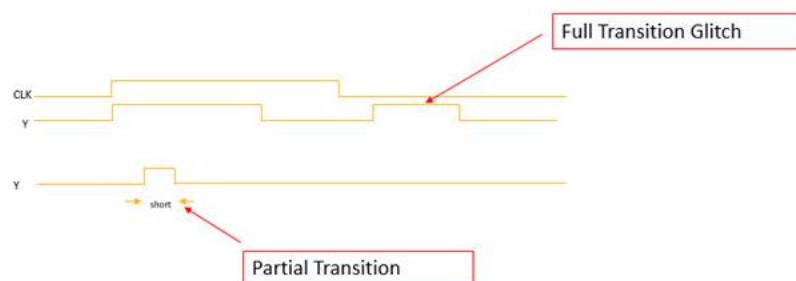


Figure 3. This example shows a common scenario of a partial transition glitch (also known as an inertia glitch, or IG) and a full transition glitch (aka a transport glitch, or TG) in a simple digital circuit

Why Is Glitch Power Important?

The following table shows the percentage of glitch power compared with total dynamic power for real-world designs. Glitch power is clearly a significant portion of the total dynamic power consumption in recent designs. Hence, this category needs to be given due attention toward optimization early in the design cycle.

Design type	Total dynamic power	Total glitch power	% glitch power
CPU1	256 mW	34.4 mW	13.44%
CPU2	314 mW	39.7 mW	12.64%
SoC1	8.93 mW	471 uW	5.27%
SoC2	64 mW	14.8 mW	23.13%
GPU	75 mW	18.9 mW	25.20%
Memory	258 mW	56 mW	21.71%
HighPerf	2.19 W	732 mW	33.42%

Table 1. Percentage of glitch power compared with total dynamic power in recent designs across multiple real-world scenarios

Why Glitch Power Is Hard to Measure

Unlike other types of power consumption in a circuit, glitch power is notoriously hard to measure with accuracy due to its nature and source. All other types of power, such as leakage power, internal power, and switching power, can be measured and estimated early in design development stages. There are well-established methods for computing other power types, either by using the circuit configuration and standard cell library information or by adding activity vector information to estimate toggles.

But glitches occur due to mismatched arrival times of converging signals. (See Figure 1.). Therefore, it is crucial to accurately measure circuit delays to identify glitches and compute glitch power. Carefully selecting process/voltage/temperature corners is also important when analyzing glitches for delay and computing power. Static timing analysis (STA) identifies only the worst-case timing through every logic stage (earliest transition and latest transition), but glitch is independent of worst-case timing. Glitch requires the actual timing of all signals, which is not provided by STA timing tools.

Considering the variability in a fabrication process, every chip, transistor, and signal will be different. This can be analyzed only with a statistical approach. But high-level statistics on glitches are not sufficient for designers to make specific design changes and reduce power.

While the source of a glitch may be difficult to determine accurately, a glitch will propagate predictably through downstream logic wherever it came from. It is possible to roughly estimate the extent of its propagation and take actions toward reducing its impact.

Another popular method of verifying power calculations at the pre-silicon stage is to compare them with power values measured post-silicon. Unfortunately, there are no known methods for calculating what portion of the post-silicon power total can be attributed to glitch power. This poses unique challenges for accurately estimating glitch power estimation at the pre-silicon stage.

Accurate, reliable glitch power estimation requires the following steps:

1. Estimate propagation delays, and use these to identify glitch toggles.
2. Compute glitch power, and annotate it on the relevant logic cells.

Trace glitch sources to pinpoint where in the circuit they originated.

Conduct Glitch Power Analysis

Keysight PowerArtist register-transfer level (RTL) power analysis and design software and the PowerArtist-SC register transfer level (RTL) design-for-power platform are designed to deliver an out-of-the-box estimate of glitch power and to identify optimizations that address the most common challenges faced by design teams today. These electronic design automation products can help take control of the increasing impact of glitch power wastage in modern designs.

The PowerArtist-SC platform is cloud-optimized for big data analytics and includes both a native delay propagation engine and a glitch identification engine. These engines can propagate delay events by reading a gate-level netlist, a timing file, timing constraints, library files, and simulation vectors. The PowerArtist-SC platform can simulate the propagation of glitchy toggles from their point of origin to their final propagated point. With this information, the tool computes the glitch power consumed by each circuit element in the propagation chain.

Calculating all this for large designs over millions of clock cycles at the gate level is compute-intensive. The PowerArtist-SC platform uses smart algorithms to minimize the computational load and distributed over hundreds of parallel central processing units. This makes it possible to deliver meaningful results to a user in a desired time.

PowerArtist software is an RTL-stage power estimation platform that includes the same glitch power engine as the PowerArtist-SC platform. This enables delivery of the same glitch power estimation and identification even earlier in the design process, at the RTL stage.

Glitch power results are shared in the same report formats as other types of power, helping designers gain insight into the glitchy parts of their design and allowing them to take corrective actions much earlier in the flow.

This extreme shift-left capability gives customers a boost in productivity and optimization that is unmatched by more traditional tools.

Reporting Glitch Power

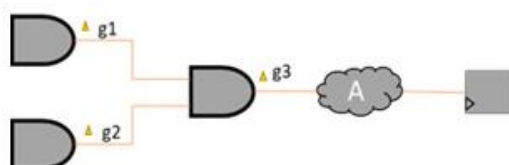
Top-level glitch power can be included in a report on average power by enabling the glitch analysis option.

1. Total power consumption

Power contribution	Power(Watts)		
	Static	Dynamic	Total
Internal power			
Internal register power	57.6uW	355uW	413uW
Internal latch power	0W	0W	0W
Internal memory power	2.61mW	1.74mW	4.35mW
Other internal power	39.8uW	29.8uW	69.6uW
Total internal power	2.71mW	2.12mW	4.83mW
Pad power	51.6uW	53.1mW	53.2mW
Clock power	2.16mW	16.8mW	19mW
Inferred Buffer power	166uW	626uW	792uW
Total glitch power	0W	2.02mW	2.02mW
Total power	5.09mW	74.7mW	79.8mW

Figure 4. A sample report showing average power, including glitch power

Glitches can propagate downstream through combinational logic, so it is important to identify instances with high self-glitch, as well as the impact on downstream logic. For example, g3 in Figure 6 is a function of both g1 and g2; fixing g1 or g2 will directly impact g3.



What is the glitch power impact of g1?

What is the glitch power impact of g2?

Fix/Reduce g1 or g2 to reduce glitch power at g3.

Figure 5. Glitch propagation in combination circuits

PowerArtist software's report on glitch sources identifies the top instances with the highest glitch impact. An instance is ranked based on the sum of both self-glitch power and downstream glitch power. For example, in Figure 7, final glitch power at g5 is a function of previously generated and propagated glitches from g1 to g4. The software is able to trace this effect back to its glitch sources and report to users in a manner that's actionable to fix the glitch sources that have the largest downstream impact.

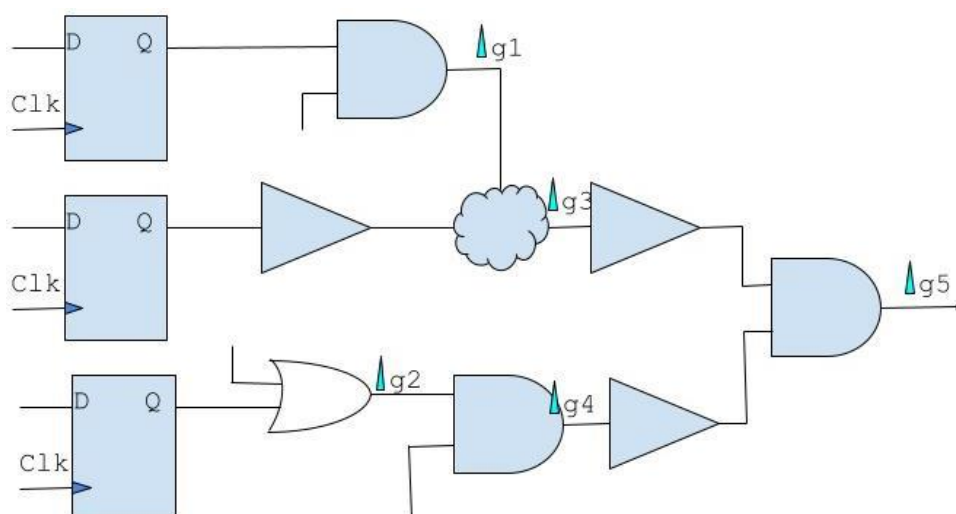


Figure 6. A sample circuit showing glitch sources and their downstream impact

Lowering Glitch Power

PowerArtist software's report on glitch sources identifies the logic with the most significant downstream impact, including self-glitches. Designers use this to examine the source code of glitchy logic and its upstream/downstream connections. With this, designers can effectively gate the identified glitch logic, conserving power and reducing waste.

This method is proven to be effective and to have a high return on investment for fixing glitches early at the RTL stage. Designers focus on the highest-impact glitch generation logic without worrying about its absolute glitch power number, given that it's too early to accurately predict this. Study shows that for a reasonable-size design, even if only the top 10 glitch sources are fixed at the RTL stage, it has a huge impact on the overall design's glitch power wastage savings down the flow.

Explore These Features

1. A simple text report lists actionable glitch sources for designers to investigate and fix.
2. Each glitch source in the RTL is reported with its own self-glitch power and its cumulative downstream impact on glitch power consumption, thus giving designers insight into the potential return on investment for fixing a glitch source.
3. To help make fixing easier, each glitch source includes its RTL cross-probing information with the RTL file name and line numbers.
4. The report intelligently combines multiple glitch sources originating from the same RTL logic to give a glitch's total impact on a design.

5. A hierarchical glitch report gives details of glitch power in different design hierarchies and highlights the main glitch-prone logic that designers should pay attention to.
6. PowerArtist software saves all glitch analysis data in its database so users can customize reports and generate deeper data analytics.

Use a Proven Solution

Glitch transitions in digital logic can be responsible for almost a third of a circuit's total power consumption, yet identifying potential glitch sources and glitch propagation are difficult to achieve with any accuracy. Despite these barriers, we have seen in this paper how PowerArtist software and the PowerArtist-SC platform can provide designers with a wealth of information on glitch power earlier in the design flow.

The products also identify the top targets for optimizing RTL design to reduce total glitch power. These measurement and optimization techniques have been developed over many years of working closely with a wide array of semiconductor companies and are available in an easy-to-use platform that delivers results out of the box. As designs grow larger, it becomes ever more important to adopt advanced shift-left techniques and take advantage of highly parallel resources available in the cloud. The PowerArtist family is the proven solution to achieve this goal.

Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.



This information is subject to change without notice. © Keysight Technologies, 2025, Published in USA, October 20, 2025, 3125-1298.EN