Journal of
Low Power Electronics and Applications

Article

A Gate-Level Power Estimation Approach with a Comprehensive Definition of Thresholds for Classification and Filtering of Inertial Glitch Pulses

Benjamin Villegas and Ioannis Vourkas *

Department of Electronic Engineering, Universidad Técnica Federico Santa Maria, Avda. España 1680, Valparaiso 2390123, Chile
* Correspondence: ioannis.vourkas@usm.cl

Abstract: Estimation of power consumption in digital circuits is performed at gate-level simulation. Its accuracy depends on the models of gate delays that capture the effects of spurious signal transitions, called "glitches". Electronic Design Automation (EDA) software considers inertial gate delays and represses a glitch in the cell's output if its width is below a threshold. Selecting threshold values for the inertial glitch classification and filtering is crucial for precise power estimations. In this direction, we explore the effectiveness of automatically adjusting such thresholds on a cell-specific basis according to the local cell's information. We used a commercial industry-standard gate-level power estimation tool and a 32 nm CMOS standard cell library. Via power measurements in circuit simulations, we created customized lookup tables for each library cell employed in the benchmark circuits. We compared the proposed approach's performance with other methods for glitch threshold definition. Our method demonstrated good power estimation accuracy while presenting the lowest mean absolute error among all the cells of the circuits under test and the smallest standard deviation. The latter suggests that the proposed method achieves better cell-specific accuracy, which is expected to allow for more precise circuit-level power estimations in complex circuits with a large number of combinational cells.

Keywords: gate-level simulation; glitch power; inertial gate delay; transport glitch; glitch filtering

check for updates

Citation: Villegas, B.; Vourkas, I. A Gate-Level Power Estimation Approach with a Comprehensive Definition of Thresholds for Classification and Filtering of Inertial Glitch Pulses. *J. Low Power Electron. Appl.* **2024**, *14*, 41. https://doi.org/10.3390/jlpea14030041

Received: 2 June 2024
Revised: 27 July 2024
Accepted: 31 July 2024
Published: 5 August 2024

1. Introduction

Power consumption has always been among the primary quality metrics in digital circuits [1]. However, the increasing demand for higher computing performance and the widespread use of portable electronic devices [2,3], have made power efficiency in electronic design a top priority in many applications [4,5]. As a result, there has been significant research interest in developing precise methods for power estimation and optimization in electronic design automation (EDA) software, as evidenced by [6–9].

Circuit simulation is the most reliable tool for calculating dynamic power. However, due to the high time complexity of circuit simulations, they may only be practical for small-scale circuit designs. Instead, the estimation of power is performed in early design stages at a more abstract level, using a more time-efficient gate-level simulation, which takes into account the switching events at the nodes of the circuit and libraries with cell-specific parameters [10–12]. Nevertheless, such speedup of calculations inevitably leads to accuracy problems. Practical gate-level power estimation is highly dependent on the development of precise models of the gate delays, which capture the effects of spurious signal transitions (partial voltage swing) [13,14]. The latter are often called "glitches" and occur at the output of logic gates due to timing issues caused by unbalanced path delays at the inputs of logic gates [15]. Glitches may trigger further switching once generated if they are propagated through subsequent levels of logic gates and thus could contribute up to 40% of total dynamic power [15–17]. Such a high contribution to power consumption demonstrates why evaluating glitches is significant for low-power circuit design [18]. Several works in the literature present research results on power estimation strategies that focus on glitch generation and propagation [19–21]. To this end, the work in [22] proposed alternative, computationally efficient glitch modeling approaches, and [23] explored strategies for glitch reduction, which is essential to avoid power-hammering attacks caused by intentional glitch amplification and excessive dynamic power consumption [24].

Evaluating the impact of glitch generation at the gate level without information about the actual input/output waveforms is challenging [14]. Given that gate-level simulation assumes only full-swing voltage transitions, power estimation for partial swing transitions, whose amplitude depends on the characteristics of the gate's input/output signals, can be based only on approximations. In this direction, probabilistic simulation techniques to estimate glitch power have been explored [20]. However, power estimation techniques in EDA software generally consider inertial gate delays and, by default, repress a glitch pulse at the output of a gate (as inertial) if its width is below a predefined classification threshold [21]. The value of the threshold used to classify a glitch as inertial in logic simulators is usually globally fixed to a certain percentage of (i.e., strictly less than or equal to) the cell's propagation delay [14,16]. In contrast, the rest of the generated glitch pulses whose width exceeds such threshold are considered transport glitches, which propagate to the output of a gate without attenuation. Moreover, there is also a filtering (elimination) threshold, below which the software drops a glitch pulse. Selecting proper threshold values for the inertial glitch pulse classification and filtering is crucial for precise power estimations. High filtering thresholds could lead to aggressive glitch filtering and thus to underestimation of dynamic power if power estimation omits many (otherwise valid) glitch pulses. On the other hand, overestimation of power could occur if classification is not appropriately applied while using a low threshold value since several glitch pulses could be propagated as transport glitches even though, in reality, they could be too narrow for this to be physically possible. So, the actual implementation of the inertial delay model can affect the accuracy of estimations for power attributed to glitch generation and propagation. Even state-of-the-art gate-level simulation software can lead to high estimation errors due to poor glitch handling [19], and such errors increase with the combinatorial logic depth in the circuits under consideration [16].

Nevertheless, gate-level simulators could achieve more comprehensive glitch power estimations by exploiting the local cell's information. More specifically, using customized lookup tables concerning different output loads and signal transition slopes, the glitch classification threshold could be properly selected to adequately capture the glitch behavior in every cell. According to [14], such a strategy can improve the accuracy of estimations, which, however, requires pre- and post-processing of simulation data and thus comes at the cost of longer runtimes. In the same context, storing extra information in the lookup tables to include characteristics of the analog waveform response, according to [19], could improve the accuracy even more, but with an arbitrarily high increase in storage requirements for the characterization library.

In this context, here we consider gate delay the only measure for inertial glitch classification and filtering. We particularly explore the effectiveness of applying automatic adjustment of the corresponding threshold values on a cell-specific basis, according to the local cell's information, as opposed to the default scenario that applies standard global classification and filtering settings for the glitch pulses. To this end, we used a commercial industry-standard gate-level power estimation tool and a 32 nm CMOS standard cell library. Via power measurements in circuit simulations, we created customized lookup tables for each library cell employed in the design of the benchmark circuits. We created the lookup tables by considering several different possible output load values and transition times of the input signals to capture output load dependency and account for changes in input signal slope while considering all primary inputs uncorrelated. Each lookup table stores the percentage of the cell's propagation delay corresponding to every threshold. We used Verilog HDL for every design to evaluate and implement using the pre-characterized standard cells. The benchmark circuits were medium-sized circuits selected from the ISCAS85 and ISCAS89 suites and custom designs, whose glitch power component ranges approximately between 8% and 20% of total dynamic power. We followed a strategy similar to [14,18] to modify the standard power estimation workflow with additional processing of the Standard Delay Format (SDF) file and posterior manual modification of the Value Change Dump (VCD) file to be able to classify glitch pulses with a width more extensive than the cell's propagation delay. Using the netlist and processed SDF file, we performed zero-delay and SDF-annotated simulations. We compared the performance of the proposed approach with that of other methods for glitch threshold definition proposed in the literature and with that used by an established gate-level power estimation tool. The results showed that the currently adopted strategies in EDA tools do not guarantee the best power estimation results. Moreover, our analysis revealed that a method's overall good estimation accuracy could be attributed only to the self-cancelation of different error contributions with opposite effects. So, its effectiveness could vary significantly depending on the specific characteristics of every design under test. Generally, the proposed method achieved good overall power estimation accuracy in several categories, outperforming the rest of the methods while presenting the lowest mean absolute error among all the cells of the circuits under test and the smallest standard deviation. The latter generally suggests that the method achieves a better cell-specific accuracy, which is expected to allow for more precise circuit-level power estimations as the number of combinational cells increases in complex circuits. Moreover, we observed positive results when transport and inertial glitch power were studied separately, where the proposed method showed a better estimation of the power attributed to transport glitches with just a 5% error.

The rest of the paper is organized as follows. The definition of glitch pulse generation and propagation is given in Section 2, along with the description of the simulation setup and some concluding analyses that reflect the dependence of glitch power on the cell's local environment. Section 3 describes the proposed approach for the proper definition of the threshold values for glitch classification and filtering on a cell-specific basis, along with the modified gate-level power estimation flow which incorporates our approach. Section 4 presents the comparison results between the proposed approach and other strategies for threshold definition, using the power measured in SPICE as a basis for evaluation. Finally, in Section 5 we discuss the major conclusions of our analyses.

2. Evaluating the Dependence of Glitch Power on the Cell's Local Environment

If zero delay could be achieved, all input signals to logic gates could switch simultaneously. As a result, the circuits would consume the least possible energy required for their operation. However, misaligned transitions at different inputs of a gate could result in the generation of glitch pulses at its output node. The circuit schematic shown in Figure 1 illustrates a typical example of glitch generation. A delay in the arrival of one of the input signals (eco_net) of the AND logic gate U1, due to the existence of a buffer in that path, results in a glitch in its output node D. This glitch is then passed on to the output node (OUT) of the next AND gate U2. Whenever glitches occur, they cause additional switching activity if they propagate through subsequent logic gates. This undesired switching can significantly increase the total energy consumption.

Established gate-level power estimation tools treat glitch pulses as inertial when their value reaches only the intermediate state (state X). Their power consumption is normally scaled by a factor $f_s < 1$. Therefore, depending on the value of f_s, the estimated power can be much less than that corresponding to a complete set of transitions ($0 \rightarrow 1 \rightarrow 0$, or vice versa). In addition, the EDA tool settings only permit glitch classification and filtering thresholds to be less than or equal to the cell's propagation delay for all cells in the circuit. Therefore, enhancing the definition of these thresholds is essential to improve the accuracy of gate-level power estimations. However, the generation and propagation of glitches vary depending on the type of cell and parameters describing its local environment. In this regard, here we will analyze the dependence of glitch power on the parameters that describe the environment of a gate. Specifically, we will observe the variations in cell performance owing to the type of logic cell, its output load, the slope of the input signals (input's transition time), and the precise sequence of signal changes that produce the glitch (signal pattern).

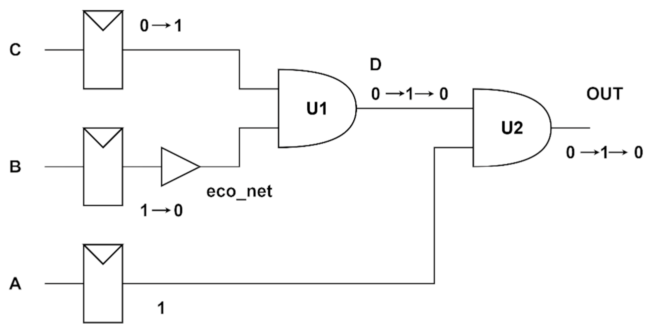


Figure 1. Circuit schematic used to exemplify glitch generation and propagation through logic cells.

In our simulation setup, we use a Python script and the relevant API to access the SPICE environment of a commercial industry-standard EDA tool to run physical-level circuit simulations and process the exported results. The overall simulation workflow is depicted in Figure 2. First, we create the input signals for the simulation in PWL format. Then, *MEAS*-type commands are created to obtain the energy corresponding to signal transitions, the cell propagation delay, as well as the amplitude and width of each pulse at the output. The measurements obtained are stored in a JSON format, which is later used to analyze the behavior of each cell. Simulations are performed for different logic cells, input patterns, output loads, and transition times of the input signals. To normalize all comparisons, we use the "*Fan-Out of Four*" (FO4) delay metric of the technology $t_{\mathrm{d,FO4}}$, and the chosen values for the transition time $\Delta t_{\mathrm{slope}}$ are arbitrarily expressed in multiples of $t_{\mathrm{d,FO4}}$, specifically between $1 \leq \Delta t_{\mathrm{slope}}/t_{\mathrm{d,FO4}} \leq 4$.

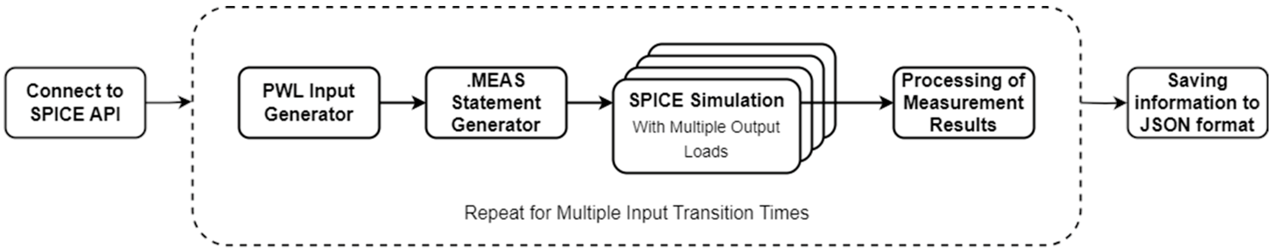

Connect to SPICE API
PWL Input Generator
.MEAS Statement Generator
SPICE Simulation
With Multiple Output Loads
Processing of Measurement Results
Saving information to JSON format
Repeat for Multiple Input Transition Times

Figure 2. Overall description of the applied circuit simulation and data processing workflow.

In Figure 3 we present simulation results for a 2-input NAND cell designed with 50 nm BSIM4 transistor models, for which it is $t_{\mathrm{d,FO4}}$ = 59.3 ps. For a specific input transition time $\Delta t_{\mathrm{slope}}$, in every simulation the cell is connected to the same output load of 1 fF and the input signals arrive with a decreasing delay between them, as shown in Figure 3a,b. The time evolution of the cell's output voltage is shown in Figure 3c, where we notice the generation of glitch pulses. Due to the decreasing delay in the arrival of the input signals as the simulation progresses, the produced glitches have different pulse widths and amplitudes. We study the power attributed to the glitches in the plot shown in Figure 3d, where we note that the maximum power remains practically the same even for shorter glitch pulses, and it is non-negligible even when the glitch pulse amplitude decreases to approximately 10% of the supply voltage V_{dd}. Next, we carried out such simulations for different cells from a 32 nm standard cell library, which are repeated for different input patterns, output loads, and transition times of the input signals, to study the evolution of the amplitude and the energy consumption corresponding to every glitch pulse, with respect to its pulse width. Our objective is to search for correlations that could lead to the development of more accurate models for cell performance. The energy consumption corresponding to every glitch pulse is calculated by integrating the instantaneous power within the time interval where power is greater than or equal to 5% of its maximum value.

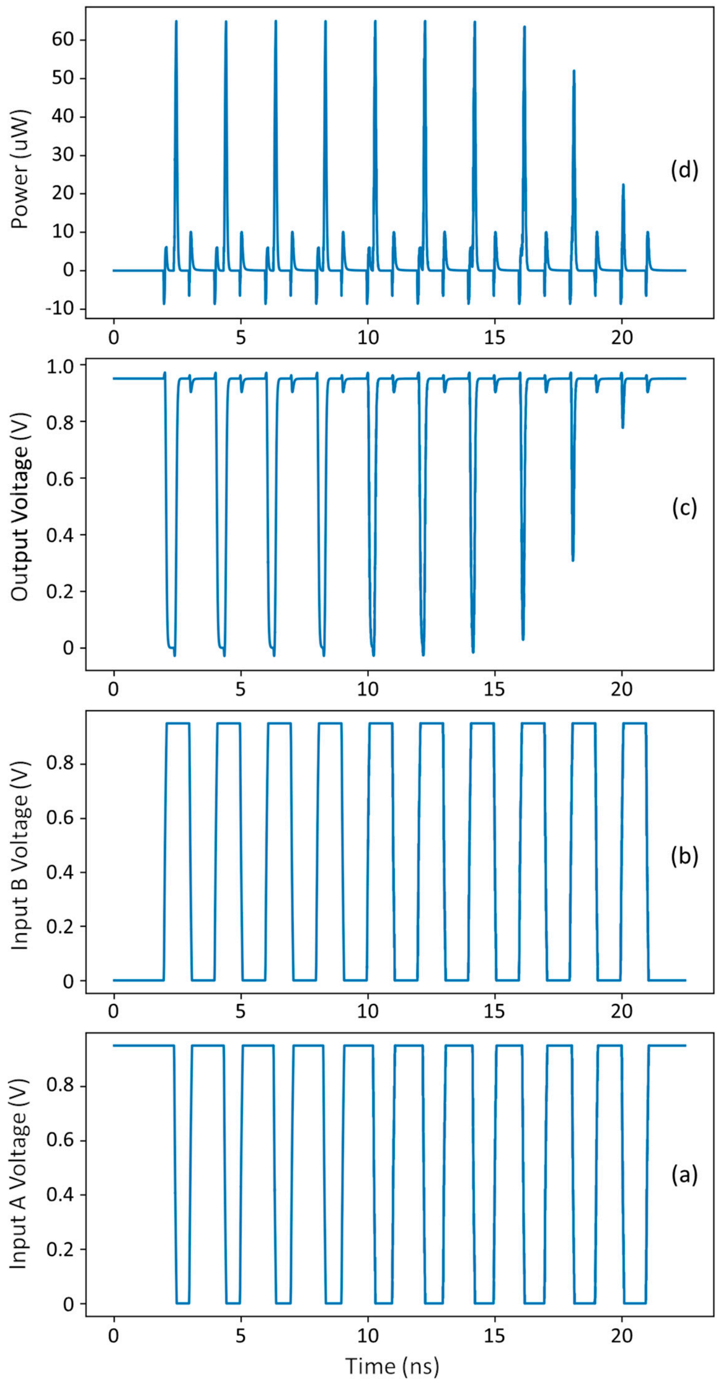
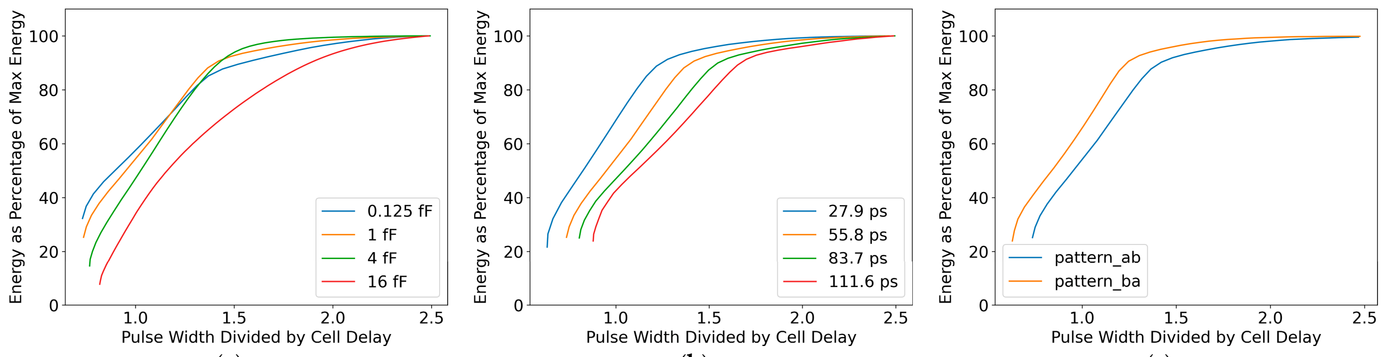


Figure 3. Simulation results for a 2-input NAND cell designed with 50 nm BSIM4 transistor models, with an output load of 1 fF. (**a**,**b**) The PWL format of the two input signals; (**c**) the time evolution of the output voltage; and (**d**) the calculated instantaneous power.

In this context, we present in Figure 4 the concluding results for the relationship between the glitch pulse width and the energy consumption for a 2-input NAND cell from a 32 nm standard cell library with $t_{d,FO4}$ = 27.9 ps, considering different local conditions. The different plots concern the variation of the output load capacitance, the input signal slope, and the order of the transitions at the input signals (input pattern). Note that the glitch pulse width Δt_{glitch} is normalized by the cell's propagation delay t_{d}, whereas the glitch energy E_{glitch} is normalized by the total energy E_{max}, which corresponds to a perfect pulse with full-swing voltage transitions.



(**a**) (**b**) (**c**)

Figure 4. Concluding results from simulations concerning a 2-input NAND cell from the 32 nm standard cell library, about the normalized glitch energy vs. normalized glitch width, for different local conditions of the cell. Plots concern: (**a**) variation of the output load capacitance; (**b**) variation of the input signal slope; and (**c**) variation of input pattern, when input B is delayed (pattern_ab) and when input A is delayed (pattern_ba).

By observing Figure 4, we note that even for $\Delta t_{\text{glitch}}/t_{\text{d}} \approx 0.75$ the measured $E_{\text{glitch}}/E_{\text{max}}$ ratio could reach approximately 40%, whereas for pulse widths closer to t_{d}, the measured E_{glitch} could exceed 60% of E_{max}. However, almost regardless of the load capacitance, E_{glitch} approximates E_{max} when $\Delta t_{\text{glitch}}/t_{\text{d}} \approx 2$. Therefore, if the inertial glitch classification threshold of an EDA tool is fixed to a value less than or equal to the cell's propagation delay, this will lead to excessive overestimation of energy consumption, unless a carefully selected value for the scale factor $f_{\text{s}} < 1$ is used. On the other hand, for $\Delta t_{\text{glitch}}/t_{\text{d}} \approx 0.5$ it seems reasonable to consider that the measured $E_{\text{glitch}}/E_{\text{max}}$ ratio becomes negligible. Such observations highlight the dependence of glitch energy on Δt_{glitch} and the relevance of proper selection of the classification and filtering thresholds. Moreover, the more abrupt the slope of the input signals, the higher the $E_{\text{glitch}}/E_{\text{max}}$ ratio, for a given $\Delta t_{\text{glitch}}/t_{\text{d}}$. Finally, we observe the influence of the input pattern, which creates different conditions for the capacitance of the internal nodes of the cell. Hence, such curves can vary significantly depending on the specific type/implementation of every logic cell.

Likewise, in Figure 5 we present the concluding results for the relationship between the glitch pulse width and the glitch pulse amplitude for the same logic cell while varying in the same manner as the local conditions. Here, the glitch pulse amplitude ΔV_{glitch} is normalized by the maximum voltage value, which is the cell's supply voltage V_{dd}. The curves follow similar trends as in Figure 4, and similar dependences are found on the local conditions. By comparing the information in equivalent plots between Figures 4 and 5, we note that E_{glitch} turns out to be non-negligible even when $\Delta V_{\text{glitch}}/V_{\text{dd}} < 0.1$. However, the slope of the curves in Figure 5 is quite steep, so the decrease in ΔV_{glitch} with respect to $\Delta t_{\text{glitch}}/t_{\text{d}}$ is fast. As a result, the very small amplitude values impact the precise detection of reference points (50% of the amplitude) in the curves, making it difficult to calculate the corresponding Δt_{glitch} when it is $\Delta t_{\text{glitch}}/t_{\text{d}} < 1$.

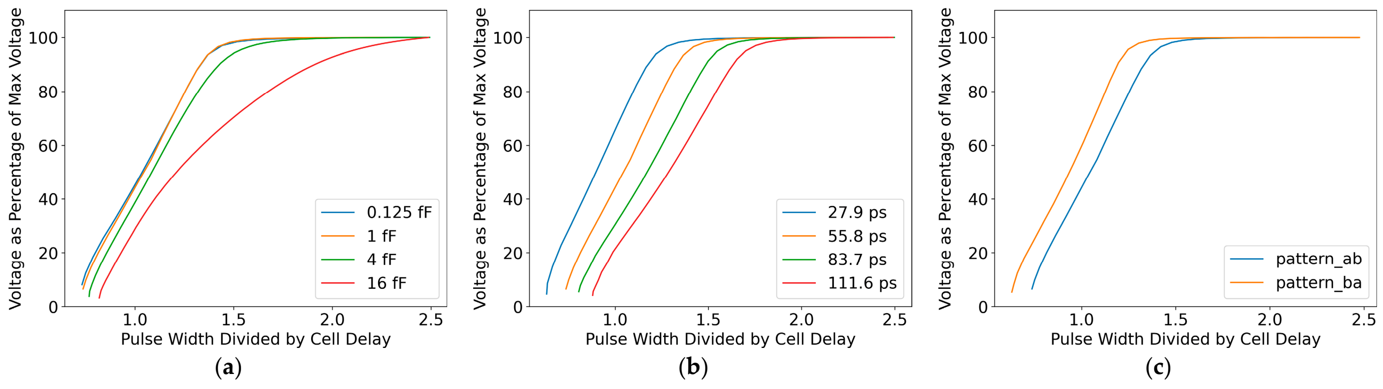


Figure 5. Concluding results from simulations concerning a 2-input NAND cell from the 32 nm standard cell library, about the normalized glitch amplitude vs. normalized glitch width, for different local conditions of the cell. Plots concern: (**a**) variation of the output load capacitance; (**b**) variation of the input signal slope; (**c**) variation of input pattern, when input B is delayed (pattern_ab) and when input A is delayed (pattern_ba).

The information presented so far indicates that the proper values to assign to thresholds for classification and elimination of glitches vary, depending on the cell's environment and the characteristics of the input signals. Moreover, the results in Figure 6 demonstrate that the performance is substantially different between different cells of the same technology, for the same local conditions. For instance, a higher $E_{\text{glitch}}/E_{\text{max}}$ ratio was obtained in the case of a 2-input XOR cell from the 32 nm standard cell library, for a given $\Delta t_{\text{glitch}}/t_{\text{d}}$ value, whereas for $\Delta t_{\text{glitch}}/t_{\text{d}} \approx 0.5$ here the measured $E_{\text{glitch}}/E_{\text{max}}$ cannot be considered negligible. Additionally, it could be more appropriate to set the threshold values higher than the cell's propagation delay to classify glitch pulses as inertial. In this direction, customized lookup tables can be generated to indicate what the appropriate threshold should be, for a variety of local conditions and possible patterns for the input signals. We explore such modeling strategies in the next sections.

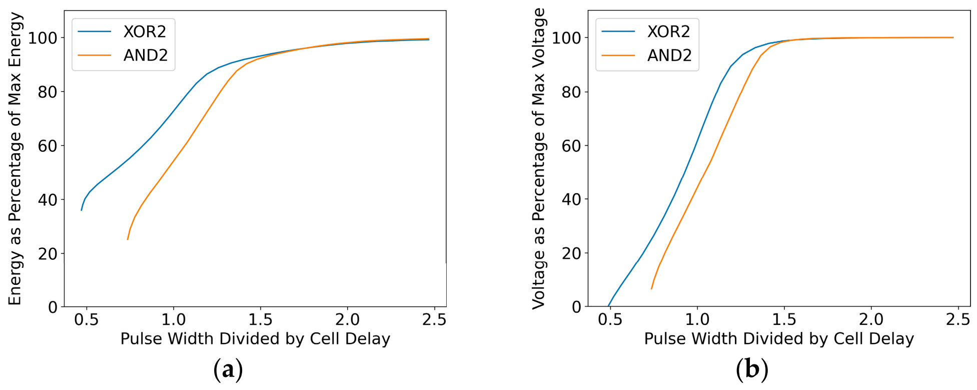


Figure 6. Comparison of simulation results concerning a 2-input NAND and a 2-input XOR cell from the same 32 nm standard cell library, about (**a**) the normalized glitch energy vs. normalized glitch width; (**b**) the normalized glitch amplitude vs. normalized glitch width, for different local conditions.

3. Proposed Method for Glitch Classification and Filtering

In this section, we explore the effectiveness of an ad hoc definition of the threshold values for glitch classification and filtering on a cell-specific basis, according to the local cell's information, as opposed to the default EDA tool settings that apply standard global classification and filtering for all logic cells. Moreover, we permit glitch classification and filtering thresholds to be assigned higher values than the cell's propagation delay. More specifically, we consider that glitch pulses should be identified as inertial when $E_{\text{glitch}}/E_{\text{max}} < th_{\text{inertial}}$, which we arbitrarily defined as $th_{\text{inertial}} = 0.95$, thus only glitches whose energy closely approximates $E_{\max}$ will be fully propagated. Likewise, glitches for which it is $E_{\text{glitch}}/E_{\max} < th_{\text{filter}}$ should be removed from the event list. In this case, we defined $th_{\text{filter}} = 0.1$, to eliminate glitch pulses only when they consume a reasonably small amount of energy, compared to $E_{\max}$. Furthermore, the effectiveness of the power estimation procedure is also studied when no glitch is filtered while using the same inertial glitch classification criterion.

Based on SPICE simulation results, we identified the normalized glitch pulse width $\Delta t_{\text{glitch}}/t_{\text{d}}$ for which the abovementioned th_{inertial} and th_{filter} thresholds are met. Next, we created customized lookup tables for all types of cells concerning different output loads, input signal slopes, and signal patterns that create a glitch. Such lookup tables store the corresponding $\Delta t_{\text{glitch}}/t_{\text{d}}$ values for classification and filtering. We present in Table 1 an example set of threshold values to classify glitch pulses as inertial in the case of a 2-input NAND cell from the 32 nm standard cell library. This table corresponds to a given input signal pattern and presents the resulting $\Delta t_{\text{glitch}}/t_{\text{d}}$ values for 32 possible combinations of output load and input transition times Δt_{slope}, which are expressed in multiples of $t_{\text{d,FO4}} = 27.9$ ps. Given that information, in power estimations, we select the precise values by interpolating the output load and input transition time of each cell in the corresponding lookup table.

Table 1. Lookup table with the inertial glitch classification threshold, expressed as $\Delta t_{\text{glitch}}/t_{\text{d}}$ ratio, for a 2-input NAND cell from the 32 nm standard cell library, for a given input signal pattern.

Load (fF) \ **Δt_{slope} (ps)**	**27.9**	**55.8**	**83.7**	**111.6**
0.125	0.799	0.926	1.053	1.156
0.25	0.783	0.905	1.028	1.126
0.5	0.760	0.868	0.986	1.080
1	0.741	0.819	0.924	1.006
2	0.761	0.800	0.866	0.931
4	0.866	0.844	0.875	0.916
8	1.111	1.040	1.014	1.017
16	1.381	1.311	1.275	1.253

The information available in the created lookup tables for each cell employed in the benchmark circuits' design, is used in a modified version of the power estimation workflow, summarized in Figure 7, based on a commercial industry-standard gate-level power estimation tool. To overcome the limitation of EDA tool settings, which assume globally applied classification and filtering thresholds for all logic cells, we incorporate additional processing of the Standard Delay Format (SDF) file to classify and filter glitches using the information in the lookup tables. However, in a post-processing step, we manually modify the Value Change Dump (VCD) file generated from the SDF annotated simulation, to be able to classify glitch pulses with a width more extensive than the cell's propagation delay. We perform zero-delay and SDF-annotated simulations, and the difference between the estimation results represents the energy contribution by creating and propagating glitches. Moreover, such value can be divided into energy dissipated due to inertial and transport glitches for further analysis of the power estimation strategy, by subtracting the energy of glitches, classified as inertial, from the total energy consumption attributed to glitches. Next, we compare the performance of the proposed power estimation approach with that of other methods for glitch threshold definition. The results for three different benchmark circuits are presented in the following section.

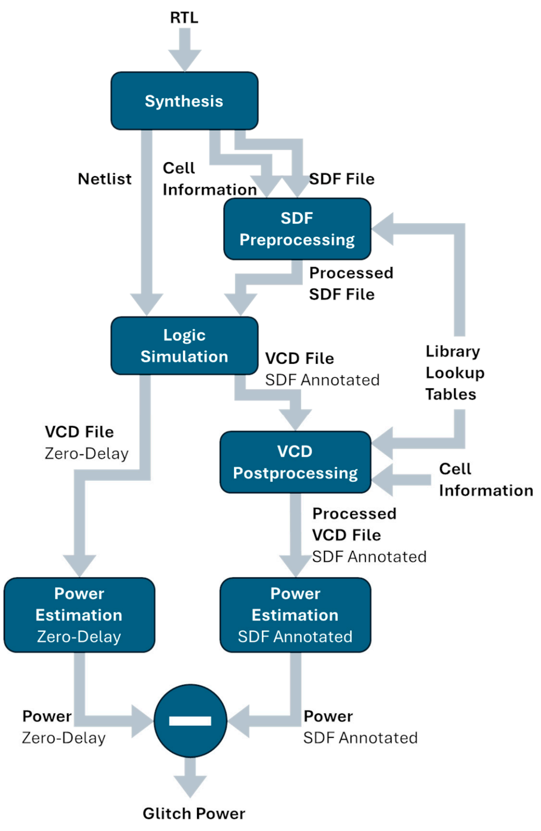


Figure 7. A modified gate-level glitch power estimation flow, including processing of the Standard Delay Format (SDF) file to classify and filter glitches based on lookup tables, and post-processing of the Value Change Dump (VCD) file to consider thresholds larger than the cell's propagation delay.

4. Results and Comparison

The analyses presented in previous sections suggest that the cells' local environment and the characteristics of the input signals affect the generation and final form of glitch pulses. Therefore, a comprehensive strategy for the definition of the classification and filtering thresholds is expected to improve the accuracy of early power estimation in EDA tools. In this direction, here we compare the proposed methodology with other strategies for threshold definition, using the power measured in SPICE as a basis for evaluation. Comparisons are made for power estimations on different circuits whose glitch power component reaches nearly up to 20% of the total dynamic power. In each case, we use the same procedure shown in Figure 7, while considering different alternatives for the threshold definition, as explained next.

The first strategy considered is the one normally adopted in the industry-standard gate-level power estimation tools, where any glitch pulse with $\Delta t_{\text{glitch}}/t_{\text{d}} < 1$ is classified as inertial. In this regard, we complement our analyses by also considering the case where no glitch is filtered, and that where any glitch pulse with $\Delta t_{\text{glitch}}/t_{\text{d}} < 1$ is directly filtered, thus without any glitch classified as inertial. The rest of the strategies were selected from the relevant literature. More specifically, in [21] a glitch is filtered when $\Delta t_{\text{glitch}}/t_{\text{d}} < 0.4$, whereas a glitch is classified as inertial when its pulse width is less than the time required for a full voltage transition $\Delta t_{\text{glitch}} < (t_{\text{rise}} + t_{\text{fall}})$. Moreover, according to [10], a glitch is filtered if $\Delta t_{\text{glitch}}/t_{\text{d,INV}} < 1.5$, where $t_{\text{d,INV}}$ is the propagation delay of a single inverter, for which we assume $t_{\text{d,INV}} = t_{\text{d,FO4}}$. Finally, the authors in [14] suggest filtering any glitch pulse that does not achieve a minimum voltage swing, for which we arbitrarily assumed a minimum voltage change (in absolute value) to be equal to 25% of the supply voltage V_{dd}. Regarding inertial glitch classification, in the absence of specific information, in the case of [10,14], we simply used the industry standard criterion of $\Delta t_{\text{glitch}}/t_{\text{d}} < 1$.

For all the benchmark circuits, simulations in SPICE concerned the application of 100 randomly generated input patterns. The first circuit under test is c17 from the ISCAS85 benchmark circuits. It consists of six two-input NAND cells, it has five inputs and two outputs, and 8% of the total power is attributed to glitches, principally inertial ones. For this circuit, 56 inertial glitches and 6 transport glitches were generated during the simulation. The second circuit under test is s27 from the ISCAS89 benchmark circuits. It is a sequential circuit that includes ten combinational cells, it has four inputs and one output, and 13.33% of the total power is attributed to glitches, principally transport ones. For this circuit, 27 inertial glitches and 23 transport glitches were generated during the simulation. Unlike c17, which has only one type of logic cell (two-input NAND gates), the s27 circuit has a variety of different combinational logic cells. Finally, the last circuit under test is a custom circuit specifically designed to guarantee a high rate for glitch pulse generation. The schematic of the circuit is shown in Figure 8. It consists of eighteen combinational logic cells, it has three inputs and one output, and 19.77% of the total power is attributed to glitches. For this circuit, 232 inertial glitches and 30 transport glitches were generated during the simulation.

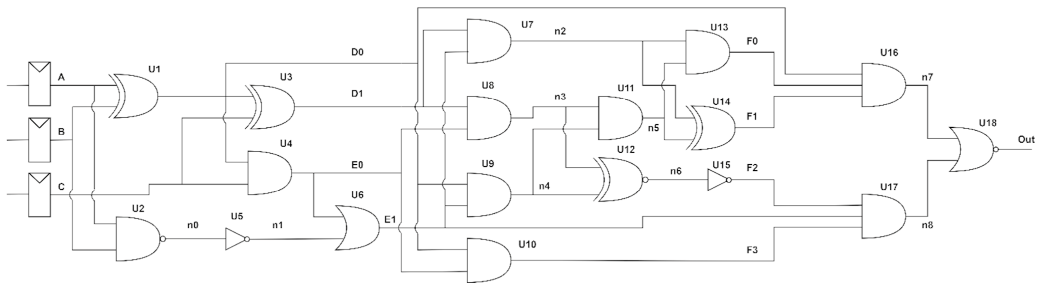


Figure 8. Schematic of the custom circuit, designed to guarantee a high generation of glitch pulses.

The comparison results for power estimation obtained for each circuit using the process of Figure 7, for all the different threshold definition strategies described above, are presented in Tables 2–4. We specifically show the estimation errors calculated for every different strategy, concerning the error in the power of the entire circuit ("*Circuit*"), but also additional metrics such as the mean absolute error of all the cells of the circuit ("*Mean Abs.*") and the standard deviation of the estimation errors among all cells of the circuit ("*Std. Dev.*"). The "*Mean Abs.*" metric is used to evaluate how far each estimate is from the real value because the "*Circuit*" column itself could lead to misinterpretations of the method's performance. In fact, a self-cancelation of different error contributions with opposite effects can result in a low total estimation error, when there are cells with large optimistic (negative) errors and others with large pessimistic (positive) errors. In every table, we identify the proposed method as "*Proposed*", the industry-standard method as "*Industry std.*", and the three alternatives from the literature, which are identified through their citation number. Note that the performance of both the "*Proposed*" and the "*Industry std.*" methods was tested also for scenarios where no filtering (*NF*) is applied to glitch pulses.

Table 2. Power estimation error for circuit c17 of the ISCAS85 benchmark circuits (NF = no filter).

	Total			**Glitches**			**Transport Glitches**			**Inertial Glitches**		
Method	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**
Industry std.	−21%	21%	5%	−60%	51%	56%	45%	21%	39%	−100%	50%	55%
Industry std. (NF)	−17%	18%	5%	−11%	36%	54%	46%	21%	39%	−32%	35%	54%

Table 2. *Cont.*

	Total			Glitches			Transport Glitches			Inertial Glitches		
Method	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**
Tsai et al. [21]	−19%	20%	4%	−36%	40%	47%	77%	21%	39%	−79%	42%	47%
Galbi et al. [10]	−20%	20%	4%	−43%	42%	48%	46%	21%	39%	−76%	41%	46%
Meixner et al. [14]	−20%	20%	4%	−43%	42%	48%	46%	21%	39%	−76%	41%	46%
Proposed	**−14%**	**16%**	**8%**	**35%**	**54%**	**93%**	**234%**	**37%**	**49%**	**−41%**	**40%**	**60%**
Proposed (NF)	−14%	16%	8%	35%	54%	93%	234%	37%	49%	−41%	40%	60%

Table 3. Power estimation error for circuit s27 of the ISCAS89 benchmark circuits (NF = no filter).

	Total			Glitches			Transport Glitches			Inertial Glitches		
Method	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**
Industry std.	−25%	25%	19%	−3%	32%	45%	41%	101%	235%	−100%	40%	52%
Industry std. (NF)	−22%	22%	18%	26%	72%	136%	40%	101%	235%	−5%	97%	154%
Tsai et al. [21]	−23%	25%	16%	13%	23%	37%	55%	65%	131%	−78%	49%	66%
Galbi et al. [10]	−25%	25%	19%	2%	27%	41%	41%	101%	235%	−83%	60%	79%
Meixner et al. [14]	−25%	25%	19%	2%	27%	41%	41%	101%	235%	−83%	60%	79%
Proposed	**−27%**	**28%**	**17%**	**−16%**	**23%**	**34%**	**5%**	**66%**	**141%**	**−62%**	**53%**	**73%**
Proposed (NF)	−24%	25%	17%	9%	70%	143%	5%	66%	141%	19%	89%	147%

Table 4. Power estimation error for the custom circuit shown in Figure 8, which had the highest number of glitch pulses (NF = no filter).

	Total			Glitches			Transport Glitches			Inertial Glitches		
Method	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**	**Circuit**	**Mean Abs.**	**Std. Dev.**
Industry std.	−18%	34%	33%	−58%	43%	60%	49%	38%	49%	−100%	44%	51%
Industry std. (NF)	−9%	29%	32%	−12%	80%	162%	23%	22%	40%	−26%	82%	133%
Tsai et al. [21]	−15%	34%	33%	−47%	44%	67%	40%	20%	38%	−81%	43%	62%
Galbi et al. [10]	−14%	32%	33%	−38%	55%	95%	23%	22%	40%	−61%	54%	75%
Meixner et al. [14]	−18%	35%	34%	−59%	43%	69%	23%	22%	40%	−91%	43%	57%
Proposed	**−15%**	**35%**	**35%**	**−45%**	**49%**	**78%**	**−7%**	**7%**	**18%**	**−61%**	**58%**	**87%**
Proposed (NF)	−10%	29%	32%	−18%	81%	138%	−7%	7%	18%	−22%	84%	141%

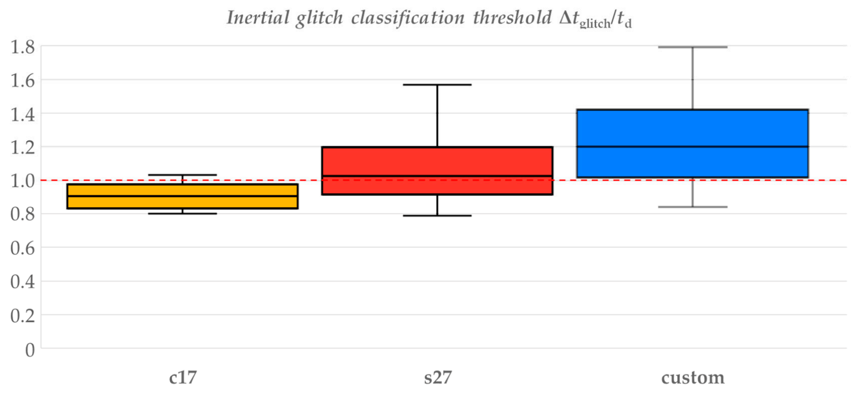
The results for circuit c17 are shown in Table 2. We observe that the estimation results for the "*Industry std.*" method present the highest total error, which underlines the need for improvement of the default glitch power estimation settings in EDA tools. Note that −100% error for inertial glitches at the circuit level occurs because all glitches with $\Delta t_{\text{glitch}}/t_{\text{d}} < 1$ are directly filtered, thus all inertial glitches are missed. All the alternative methods from the literature demonstrate a similar performance. Notably, the "*Proposed*" method presents the lowest total power estimation error, with the lowest "*Mean Abs.*" value, even though it presents a large overestimation in transport glitch power, mainly for the output cells of the circuit, which could be attributed to the higher capacitive load. In fact, if the output cells are not considered, the error in transport glitch power decreases to 13%. The thresholds defined by the "*Proposed*" method took values between $0.80 \leq \Delta t_{\text{glitch}}/t_{\text{d}} \leq 1.03$

in the corresponding lookup tables, thus in line with the industry standard criterion of $\Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} < 1$. Regarding the power of inertial glitches, the "*Proposed*" method shows nearly the best results, but when the error is evaluated on a cell-by-cell basis through the "*Mean Abs.*" metric, the other methods achieve a similar accuracy with the "*Proposed*" method.

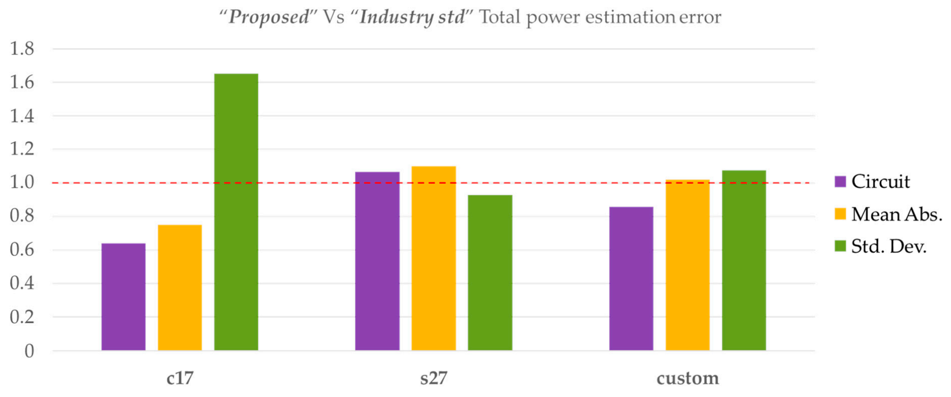
Table 3 shows the results for circuit s27, which had different combinational cells, compared to circuit c17, and the number of generated inertial and transport glitches was more balanced. Here, glitch power estimates are closer to their real value. For the "*Proposed*" method, the inertial glitch classification threshold varies between $0.79 \leq \Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} \leq 1.57$ in the corresponding lookup tables, thus taking values higher than the industry standard criterion of $\Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} < 1$. On the other hand, the glitch filtering threshold was $\Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} \leq 0.48$, thus substantially below the industry standard criterion. We observe a wider range of values here compared to the previous circuit due to the variety of existing logic cells. Here, the "*Proposed*" method does not provide the best estimate neither for the total power or the glitch power. However, it presents the lowest "*Mean Abs.*" and "*Std. Dev.*" values for the glitch power, which generally suggests that the method achieves a better cell-specific accuracy, which is expected to lead to better circuit-level estimations as the number of combinational cells increases. Moreover, positive results are observed when the transport and the inertial glitch power are studied separately. In fact, the estimation error is relatively low, achieving only a 5% error in power attributed to transport glitches. It is worth noting that other methods' overall good estimation accuracy, including the "*Industry std.*" method, is often attributed to the self-cancelation of different error contributions with opposite effects. For instance, we notice a lower error in glitch power because of the pessimistic error for transport glitches and the high optimistic error for inertial glitches in Galbi et al. [10]. This is explained by evaluating the "*Mean Abs.*" and "*Std. Dev.*" values, where the "*Proposed*" method shows good results. Table 3 also reveals the problem that NF approaches generally have, since they present the largest cell-to-cell errors, as evidenced by their "*Std. Dev.*" values.

Finally, Table 4 shows the results for the custom circuit, which generated the largest number of glitches in simulation, principally inertial ones. For this circuit, all methods under test conditions achieved similar accuracy in the total power estimation, as demonstrated by the "*Mean Abs.*" and "*Std. Dev.*" values. However, the "*Industry std.*" method with filtering presents the highest estimation errors. On the other hand, the "*Proposed*" method achieved the third best performance in total power. However, specifically in transport glitch power the "*Proposed*" method achieved by far the best accuracy. This time, the inertial glitch classification threshold varied between $0.84 \leq \Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} \leq 1.79$ in the corresponding lookup tables, thus taking values again considerably higher than the industry standard criterion of $\Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} < 1$. On the other hand, the glitch filtering threshold was $\Delta t_{\mathrm{glitch}}/t_{\mathrm{d}} \leq 0.69$. Compared to the values applied in the previous circuits, here the maximum values assigned to the thresholds increased. A high inertial classification threshold contributed to the best identification of transport glitches. We again observe positive results when the transport and the inertial glitch power are studied separately, where the estimation error was only 7% in power attributed to transport glitches. However, due to the relatively high filtering threshold values, many glitch pulses might have been erroneously filtered. In fact, when no filtering (NF) was applied, the "*Proposed*" method outperformed the rest in almost all categories, but at the cost of a poor cell-specific estimation, as demonstrated by the higher "*Mean Abs.*" value.

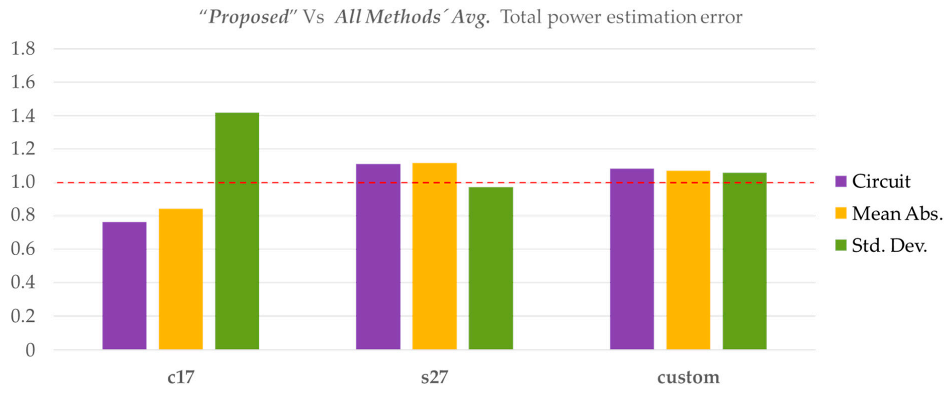
A different representation of our results is shown in Figure 9 to facilitate the evaluation of the performance of the "*Proposed*" method. For each benchmark circuit, we present the statistics of the inertial glitch classification threshold assumed by the "*Proposed*" method. Moreover, we present the total power estimation error demonstrated by the "*Proposed*" method compared with the "*Industry std.*" error and the average error of the rest of the methods. As the circuit complexity increases, the range of values for the inertial glitch classification threshold also increases beyond the industry standard criterion. The performance of the "*Proposed*" method is kept similar or better than the "*Industry std.*" method in increasingly complex circuits, with similar accuracy shown by the "*Mean Abs*." and "*Std. Dev*." values. Additionally, for larger circuits, it is shown that the "*Proposed*" method reaches a very similar average performance to the other methods considered in this study.



(**a**)



(**b**)



(**c**)

Figure 9. (**a**) Statistics for the inertial glitch classification threshold assumed by the "*Proposed*" method in every benchmark circuit. (**b**) Total power estimation error of the "*Proposed*" method, divided by the "*Industry std.*" error, which is used as reference. (**c**) Total power estimation error of the "*Proposed*" method, divided by the average error of the rest of the methods in Tables 2–4, which is used as reference. In all cases, the horizontal red dashed line highlights the reference value.

5. Conclusions

To improve the accuracy of gate-level power estimations in Electronic Design Automation (EDA) tools, this work explored the effectiveness of a more comprehensive definition of the inertial glitch classification and filtering thresholds on a cell-specific basis using customized lookup tables, prepared to properly reflect the glitch formation according to the local cell's information. The proposed approach was evaluated against different methods for threshold definition, using a modified gate-level power estimation flow and benchmark circuits designed with a 32 nm CMOS standard cell library. Our analyses highlighted several limitations in the approaches normally adopted by the industry-standard EDA tools. For instance, global tool settings only permit glitch classification and filtering thresholds to be less than or equal to the cell's propagation delay ($\Delta t_{\text{glitch}}/t_{\text{d}} \leq 1$) for all the cells of the circuit. However, the presented results showed that power estimations could be improved by selecting threshold values where $\Delta t_{\text{glitch}}/t_{\text{d}} > 1$. In fact, transport glitches could be better captured with higher inertial classification thresholds. However, high filtering threshold values could lead to excessive elimination of valid inertial glitch pulses and underestimation of glitch power. The original hypothesis for transport glitches when $E_{\text{glitch}}/E_{\text{max}} \geq 0.95$ turned out to be correct for the proposed method, but the filtering target of $E_{\text{glitch}}/E_{\text{max}} < 0.1$ could be too high, considering that the corresponding $\Delta t_{\text{glitch}}/t_{\text{d}} >$ value when this energy ratio was met reached up to 0.69. Therefore, smaller filtering threshold values could possibly improve power estimations and benefit the identification of inertial glitch pulses.

Furthermore, our detailed analyses took into consideration additional metrics, such as the "*Mean Abs.*" and "*Std. Dev.*" values, to facilitate the detection of cases where a method's overall good estimation accuracy in the total error is simply attributed to the self-cancelation of different error contributions with opposite effects. Generally, our comparisons showed that the currently adopted strategies in EDA tools do not guarantee the best power estimation results, which is what practically promotes the exploration of alternative techniques, as in this work. Moreover, since glitch power is known to be normally scaled by a factor $f_{\text{s}} < 1$ by gate-level power estimation tools, further exploration is needed to determine the proper use of such a parameter, given that results on inertial glitch power were not as good as expected.

Generally, the proposed method achieved good overall power estimation accuracy in several categories, including for glitch power, sometimes outperforming the rest of the methods with the lowest "*Mean Abs.*" and "*Std. Dev.*" values. The latter generally suggests that the method achieves a better cell-specific accuracy, which is expected to lead to more precise circuit-level power estimations as the number of combinational cells increases in more complex circuits. Note also that the proposed method, specifically in transport glitch power, achieved by far the best accuracy in two of the three evaluated circuits.

Author Contributions: Conceptualization, B.V.; methodology, B.V.; software, B.V.; validation, B.V. and I.V.; formal analysis, B.V.; investigation, I.V. and B.V.; resources, B.V.; data curation, B.V.; writing—original draft preparation, I.V. and B.V.; writing—review and editing, I.V.; visualization, I.V. and B.V.; supervision, I.V.; project administration, I.V.; funding acquisition, I.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Chilean government under grants FONDECYT Regular No. 1221747 and ANID-Basal FB0008.

Data Availability Statement: No data are available online. The data that support the findings of this study can be available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nasser, Y.; Lorandel, J.; Prévotet, J.-C.; Hélard, M. RTL to Transistor Level Power Modeling and Estimation Techniques for FPGA and ASIC: A Survey. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *40*, 479–493. [CrossRef]
2. Richa, M.; Prévotet, J.-C.; Dardaillon, M.; Mroué, M.; Samhat, A.E. High-level power estimation techniques in embedded systems hardware: An overview. *J. Supercomput.* **2023**, *79*, 3771–3790. [CrossRef]
3. Alioto, M. (Ed.) *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*; Springer: Cham, Switzerland, 2017. [CrossRef]
4. Morgenshtein, A. Short-Circuit Power Reduction by Using High-Threshold Transistors. *J. Low Power Electron. Appl.* **2012**, *2*, 69–78. [CrossRef]

5. What Is Low Power Design? Available online: https://www.synopsys.com/glossary/what-is-low-power-design.html (accessed on 1 June 2024).
6. Dinh, Q.; Chen, D.; Wong, M.D.F. Dynamic power estimation for deep submicron circuits with process variation. In Proceedings of the 2010 Asia and South Pacific Design Automation Conference (ASP-DAC), Taipei, Taiwan, 18–21 January 2010.
7. Zhang, Y.; Hu, X.; Feng, X.; Hu, Y.; Tang, X. An Analysis of Power Dissipation Analysis and Power Dissipation optimization Methods in Digital Chip Layout Design. In Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019.
8. Zou, S.; Zhang, J.; Shi, B.; Luo, G. PowerSyn: A Logic Synthesis Framework With Early Power Optimization. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 203–216. [CrossRef]
9. Nasser, Y.; Sau, C.; Prévotet, J.-C.; Fanni, T.; Palumbo, F.; Hélard, M.; Raffo, L. NeuPow: A CAD Methodology for High-level Power Estimation Based on Machine Learning. *ACM Trans. Des. Autom. Electron. Syst.* **2020**, *25*, 1–29. [CrossRef]
10. Galbi, D.E.; Kannan, K.K. Measuring Active Power Using pt px a User Perspective. In Proceedings of the Synopsys Users Group Conference (SNUG), Boston; 2010. Available online: https://veripool.org/papers/Active_Power_Primetime_PX_SNUGBos10_paper.pdf (accessed on 1 June 2024).
11. Zhang, Y.; Ren, H.; Sridharan, A.; Khailany, B. GATSPI: GPU Accelerated Gate-Level Simulation for Power Improvement. In Proceedings of the 2022 ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 10–14 July 2022.
12. Ramesh, S.R.; Jayaparvathy, R. Toggle rate estimation and glitch analysis on logic circuits. In Proceedings of the 2017 IEEE International Workshop On Integrated Power Packaging (IWIPP), Delft, The Netherlands, 5–7 April 2017.
13. Israsena, P.; Summerfield, S. Novel pattern-based power estimation tool with accurate glitch modeling. In Proceedings of the 2000 IEEE International Symposium on Circuits and Systems (ISCAS), Geneva, Switzerland, 28–31 May 2000.
14. Meixner, M.; Noll, T.G. Limits of gate-level power estimation considering real delay effects and glitches. In Proceedings of the 2014 International Symposium on System-on-Chip (SoC), Tampere, Finland, 28–29 October 2014.
15. What Is Glitch Power? Available online: https://www.synopsys.com/ai/what-is-glitch-power.html (accessed on 1 June 2024).
16. Favalli, M.; Benini, L. Analysis of glitch power dissipation in CMOS ICs. In Proceedings of the 1995 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Newport Beach, CA, USA, 23–26 April 1995.
17. Rabe, D.; Nebel, W. Short circuit power consumption of glitches. In Proceedings of the 1996 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Monterey, CA, USA, 12–14 August 1996.
18. Liu, Z.; Yu, N.; Si, H. Research on Glitch Power Estimation of Combinational Logic Circuits. In Proceedings of the 2022 International Conference on Integrated Circuits and Microsystems (ICICM), Xi'an, China, 28–31 October 2022.
19. Meixner, M.; Noll, T.G. Accurate Estimation of CMOS Power Consumption Considering Glitches by Using Waveform Lookup. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 787–791. [CrossRef]
20. Wang, L.; Olbrich, M.; Barke, E.; Büchner, T.; Bühler, M. Fast dynamic power estimation considering glitch filtering. In Proceedings of the 2009 IEEE International SOC Conference (SOCC), Belfast, UK, 9–11 September 2009.
21. Tsai, W.-C.; Shung, C.B.; Wang, D.C. Accurate logic-level power simulation using glitch filtering and estimation. In Proceedings of the 1996 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS), Seoul, Republic of Korea, 18–21 November 1996.
22. Karafyllidis, I.; Mavridis, S.; Soudris, D.; Thanailakis, A. Estimation of power dissipation in glitching using complex-time cellular automata. In Proceedings of the 1999 IEEE International Conference on Electronics, Circuits and Systems (ICECS), Paphos, Cyprus, 5–8 September 1999.
23. Shum, W.; Anderson, J.H. FPGA glitch power analysis and reduction. In Proceedings of the 2011 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Fukuoka, Japan, 1–3 August 2011.
24. Matas, K.; La, T.M.; Pham, K.D.; Koch, D. Power-hammering through Glitch Amplification—Attacks and Mitigation. In Proceedings of the 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Fayetteville, AR, USA, 3–6 May 2020.