SMART-GPO: Gate-Level Sensitivity Measurement with Accurate Estimation for Glitch Power Optimization

Yikang Ouyang[1], Yuchao Wu[1], Dongsheng Zuo[1], Subhendu Roy[3], Tinghuan Chen[4], Zhiyao Xie[2], Yuzhe Ma[1]

Presenter: Yikang Ouyang

youyang929@connect.hkust-gz.edu.cn

Jan. 23, 2025

1 The Hong Kong University of Science and Technology (Guangzhou)

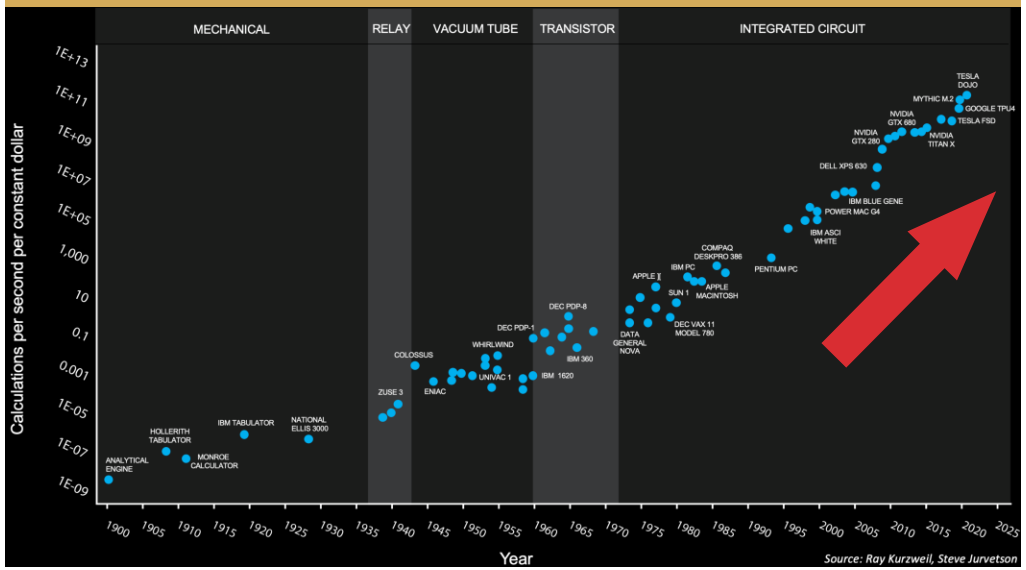
2 The Hong Kong University of Science and Technology

3 Cadence Design Systems

4 The Chinese University of Hong Kong, Shenzhen

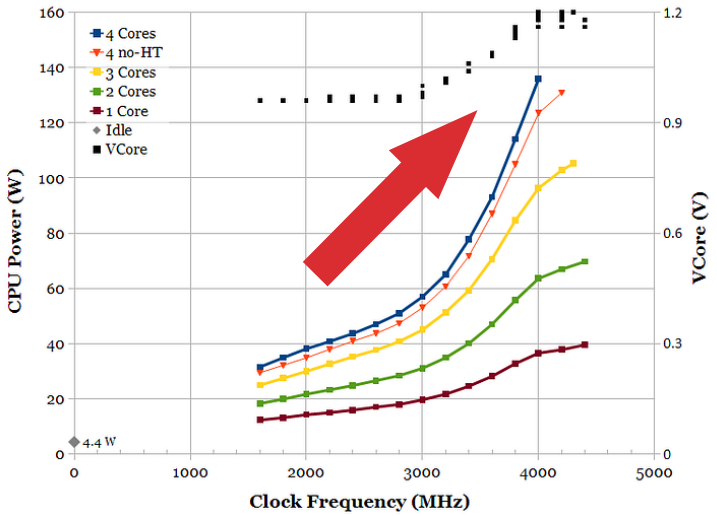
Power Consumption Issue

Increasing computation of ICs



Source: Ray Kurzweil

Power consumption



Implications

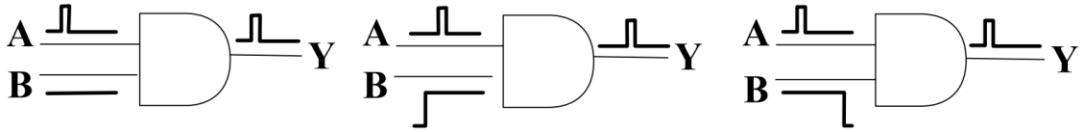
- Cooling
- Reliability
- Power delivery
- Battery life
- Carbon footprint
- …

VLSI design is hitting "power wall" ➡ **Power becomes the key objective**

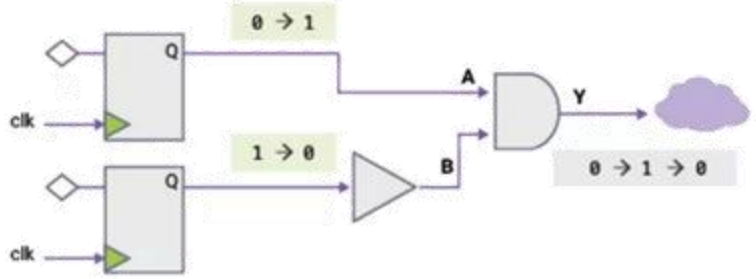
Glitch

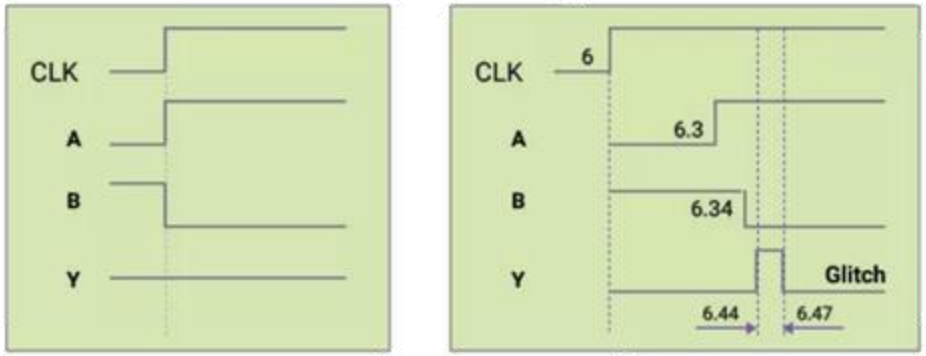
- Glitches are extra toggles due to the arrival time imbalances and signal toggling in the netlist.
- Extra dynamic power consumed

Glitch generation



Glitch propagation





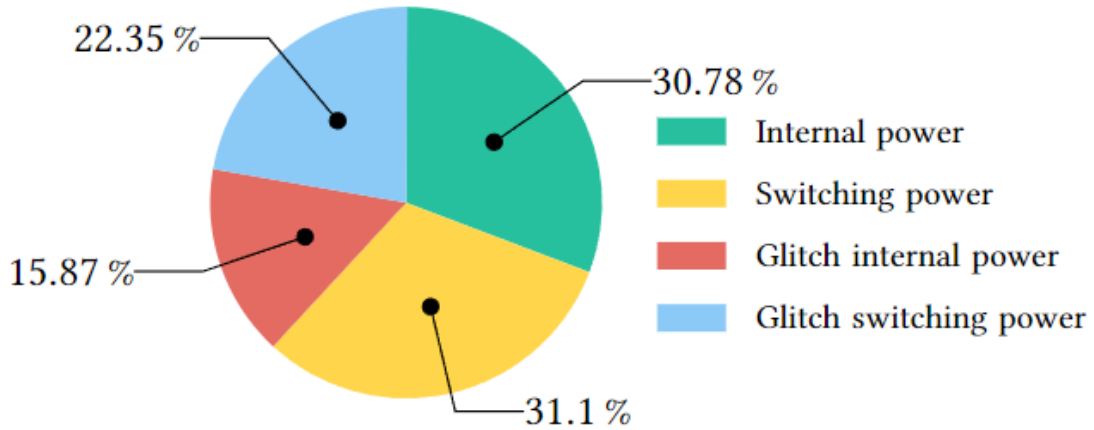
Source: Synopsys

https://www.synopsys.com/glossary/what-is-glitch-power.html#:~:text=Definition,of%20additional%20dynamic%20power%20consumption.

Glitch Power Issues

- Cause up to ***40%*** of additional dynamic power consumption
- Proportional to the number of operations

Circuit	% Glitch
alu4	25.7
apex2	29.2
apex4	30.3
bigkey	29.6
clma	24.2
des	45.4
diffeq	5.8
dsip	29.9
elliptic	12.2
ex1010	35.0



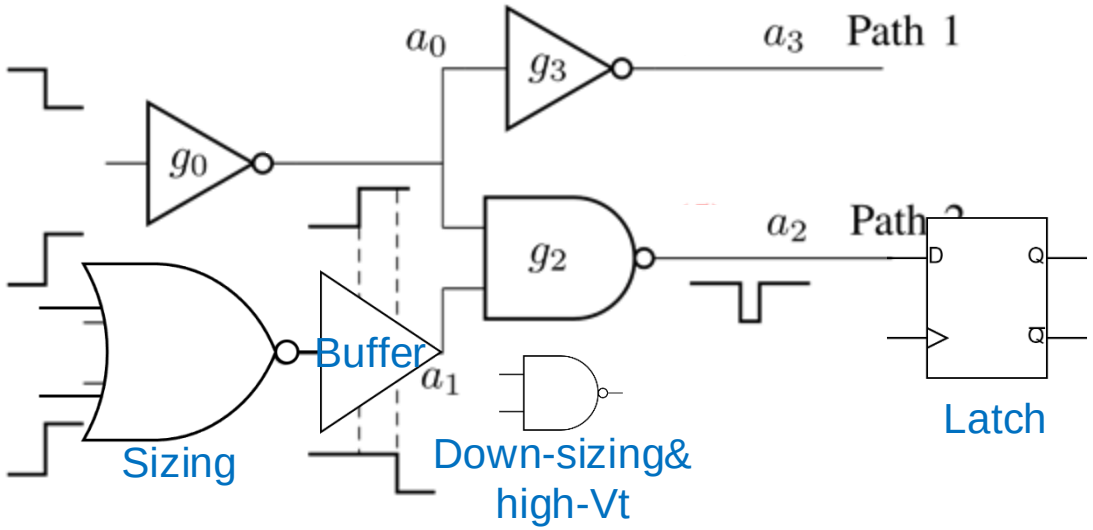
Power profiling on a BOOM chip

Source: Synopsys

https://www.synopsys.com/glossary/what-is-glitch-power.html#:~:text=Definition,of%20additional%20dynamic%20power%20consumption.

Glitch Power Optimization: ECO

- Filter a propagated glitch: latch insertion…
- Balance the input arrival time: sizing, buffering…
- Increase the inertial delay: down-sizing, use high-Vt cells…



Source: [Vithagan+, TCAD'23]

Overhead: area, power, timing, more glitch

Related Works

- Geometric programming (Global)

$$\min \sum_{c \in C} \sum_{m \neq n} \left(\frac{a_n}{a_m}\right)^2$$

$$\text{subject to } a_m \leq a_n$$

$$a_l + g_i\left(\frac{\sum_{x_j \in O_i} g_j x_j}{g_i x_i}\right) + p_i \leq a_i$$

$$a_k \leq T_{\text{spec}}$$

$$\sum_{i \in G} x_i \leq k_2 \times A_0$$

$$x_i \leq x_{\max}$$

$$x_i \geq 1$$

$$\forall\, a_l \in A_i \;\; \forall\, i \in G$$

$$\forall\, k \in P_O \;\forall\, a_m, a_n \in A_c$$

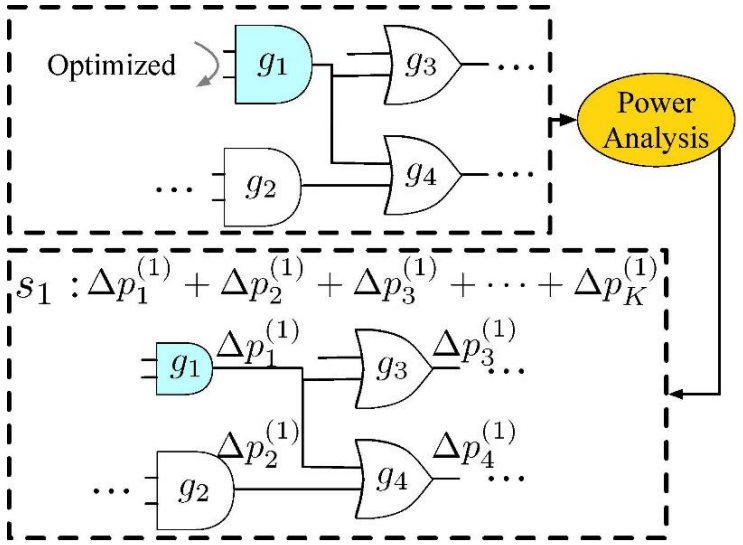
[Vithagan+, TCAD'23]

- Heuristic search (Local)
  - Glitch generation and propagation
  - Load capacitance
  - Rank gates for opt.
  - Glitch criticality [Bathla+, TVLSI'19]
  - Power metric [Wang+, SOCC'11]
    - No guarantee on result.
    - Any systematic framework?

K. Muthamizh Vithagan, V. Sundaresha, and J. Viraraghavan, "Geometric Programming Approach to Glitch Minimization via Gate Sizing," IEEE TCAD, 2023
S. Bathla, R. M. Rao, and N. Chandrachoodan, "A Simulation-Based Metric to Guide Glitch Power Reduction in Digital Circuits," IEEE TVLSI, 2019.
L. Wang, M. Olbrich, E. Barke, T. Büchner, M. Bühler, and P. Panitz, "A gate sizing method for glitch power reduction," in Proc. SOCC, 2011.

Sensitivity Measurement

- Optimizing a gate may cause glitch power changes on other gates
  - Downsize and assign high-Vt
- *Sensitivity matrix S* for global analysis
  - $\Delta p_i^{(k)}$: glitch power change on gate i after optimizing gate k

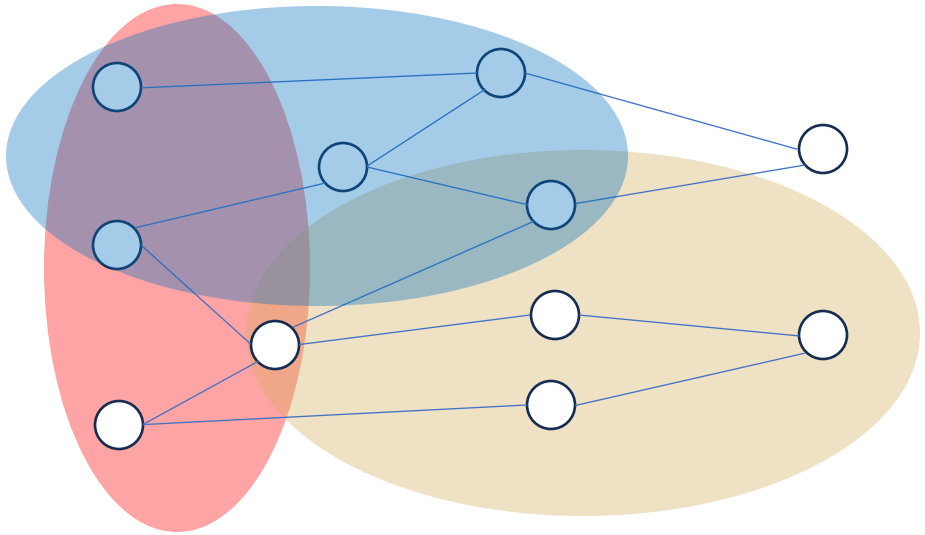


$$S = \begin{bmatrix} \Delta p_1^{(1)} & \Delta p_1^{(2)} & \cdots & \Delta p_1^{(K)} \\ \Delta p_2^{(1)} & \Delta p_2^{(2)} & \cdots & \Delta p_2^{(K)} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta p_K^{(1)} & \Delta p_K^{(2)} & \cdots & \Delta p_K^{(K)} \end{bmatrix}$$

- Per-gate sensitivity: $s_k = P^{(k)} - P = \sum_{i=1}^{K} p_i^{(k)} - \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \Delta p_i^{(k)}$

Sensitivity Measurement

- Option 1: exhaustively evaluate the impact of each gate
  - Not possible for large designs
- Option 2: sampling + statistical estimation
  - $x_i^{(b)} \in \{0, 1\}$, whether gate i is sampled in batch b
  - $\Delta P^{(b)}$: total glitch power change after optimizing the gates in batch b



Batch 1: $\left[x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}\right] \rightarrow \Delta P^{(1)}$

Batch 2: $\left[x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}\right] \rightarrow \Delta P^{(2)}$

Batch 3: $\left[x_1^{(3)}, x_2^{(3)}, \ldots, x_n^{(3)}\right] \rightarrow \Delta P^{(3)}$

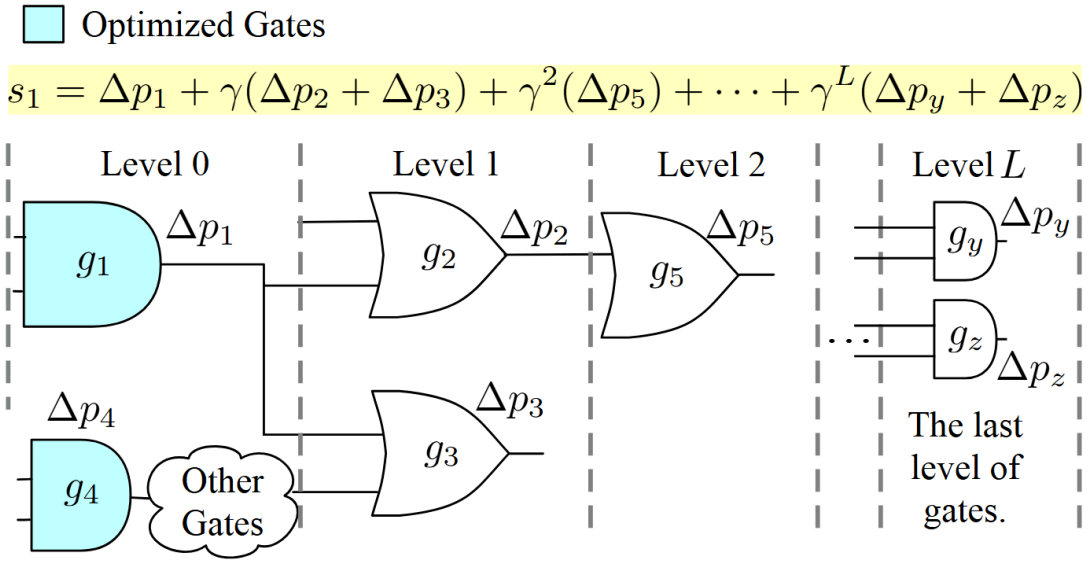
……

Batch B: $\left[x_1^{(B)}, x_2^{(B)}, \ldots, x_n^{(B)}\right] \rightarrow \Delta P^{(B)}$

$$s_i = \frac{\sum_{b=1}^{B} \Delta P^{(b)} x_i^{(b)}}{\sum_{b=1}^{B} x_i^{(b)} + \epsilon}$$

Fine-grained Sensitivity Computation

- Identify more fine-grained impact of each gate to differentiate gate sensitivity in a batch.
- Principle:
  1. Only count the power change in the fanout cone
  2. The longer the distance, the smaller the impact



$$s_i = \frac{\sum_{b=1}^{B} \Delta P^{(b)} x_i^{(b)}}{\sum_{b=1}^{B} x_i^{(b)} + \epsilon}$$

$$s_i = \frac{\sum_{b=1}^{B} \sum_{k=1}^{n} w_{i,k} \Delta p_k^{(b)} \, x_i^{(b)}}{\sum_{b=1}^{B} x_i^{(b)} + \epsilon}$$

How to Sample?

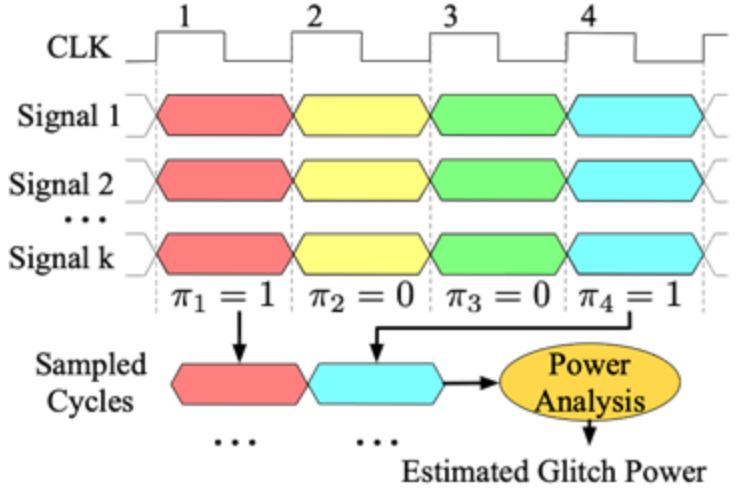
- Effective? Need more samples
- Efficient? Select “useful” samples
- Guidance for sampling?

$$\mathcal{P}(g_k) = \frac{m_k}{\sum_{i=1}^{K} m_i}$$

- Any user-defined metrics can apply, e.g.,
  - per-gate glitch power
  - power metric [Wang+, SOCC’11]
  - glitch criticality metric [Bathla+, TVLSI’19] …
- A systematic framework to analyze and optimize glitch power.

Fast Glitch Power Analysis

- Previous works (inaccurate) :
  - glitch propagation + simple timing model [Najm, TCAD'93] [Wang+, ICCAD'11]
- Golden result (slow):
  - gate-level simulation to dump traces
  - power analysis based on simulation traces → >10 hours on RISC-V BOOM
- Fast analysis: based on sampled cycles instead of full cycles → >10X faster

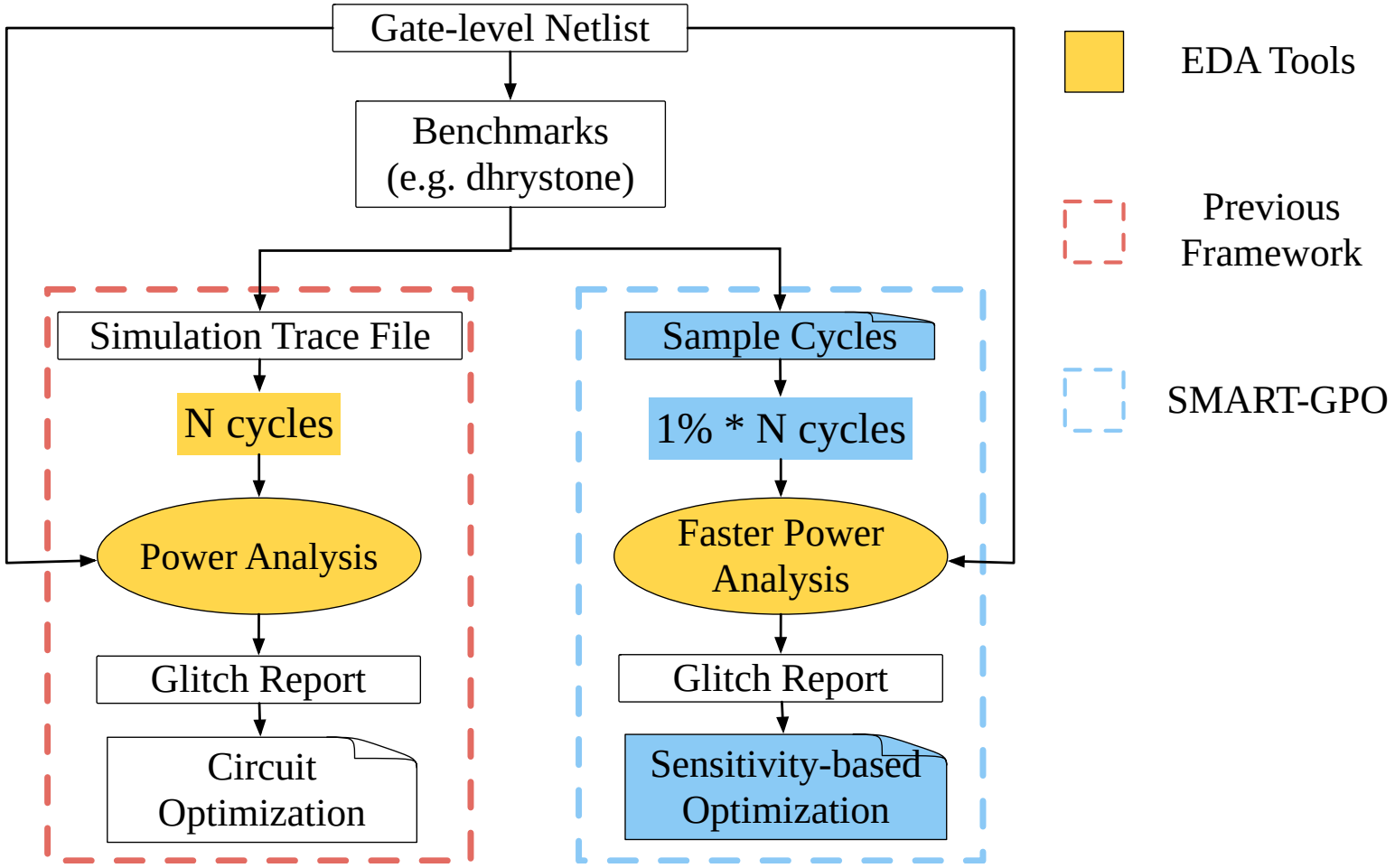


F. Najm, "Transition density: a new measure of activity in digital circuits," IEEE TCAD, Feb. 1993

L. Wang, M. Olbrich, E. Barke, T. Büchner, M. Bühler, and P. Panitz, "A theoretical probabilistic simulation framework for dynamic power estimation", Proc.ICCAD. 2011

Framework

Fast Estimation + Efficient Optimization

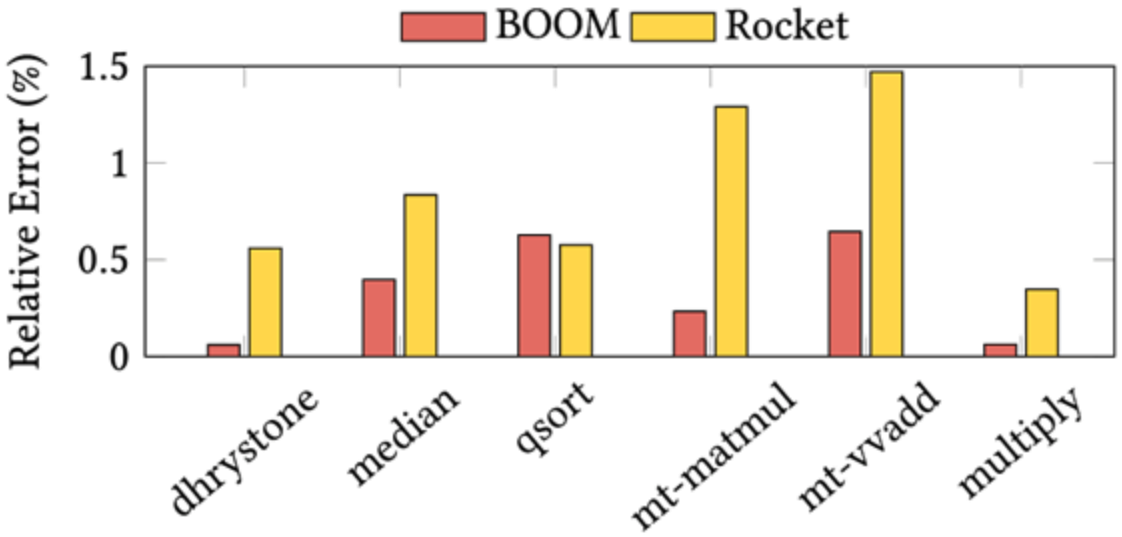


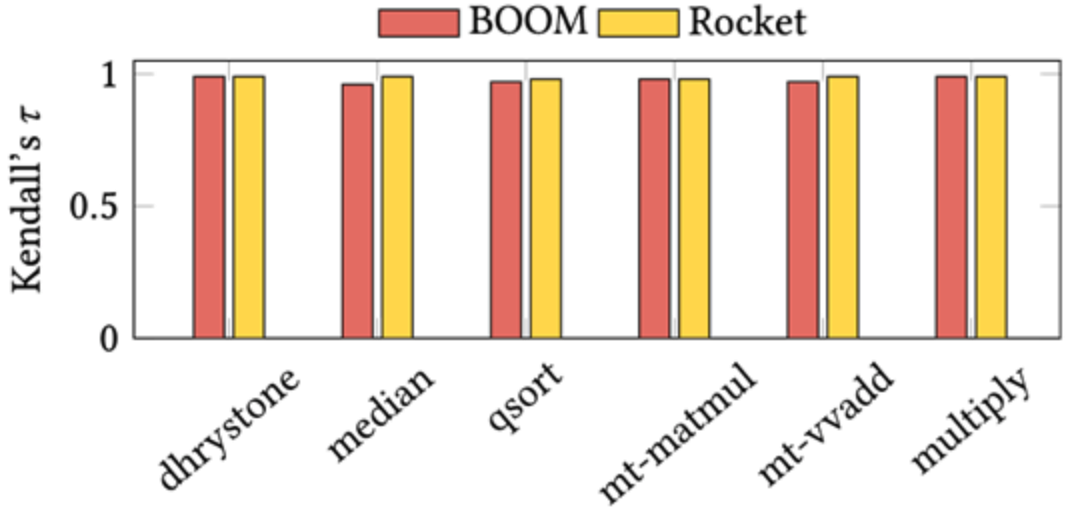
Experimental Setups

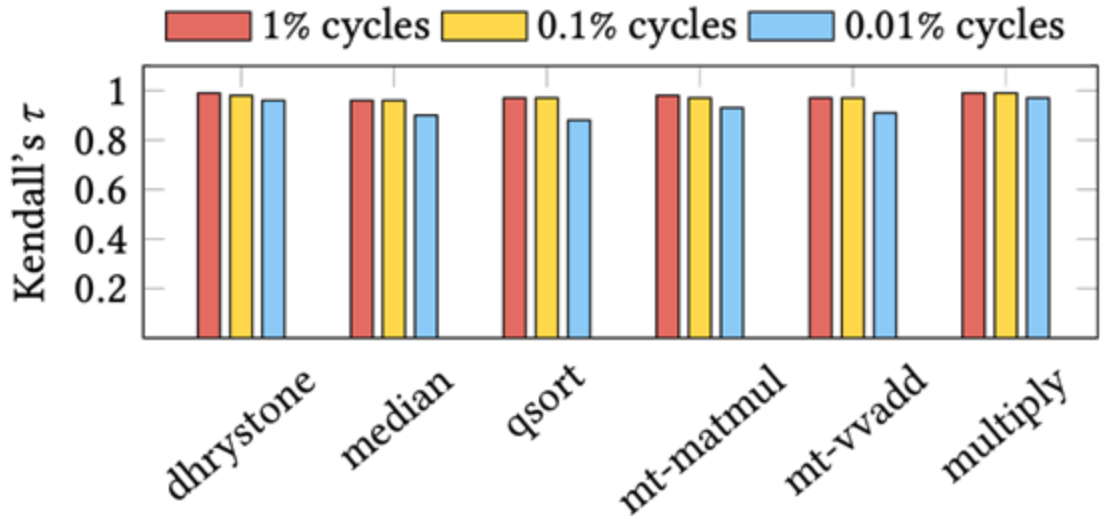
- Circuit: RISC-V designs, BOOM and Rocket
- Process: TSMC N28
- Design flow:
  - logic synthesis + gate-level simulation + time-based power analysis
- Workloads: 6 benchmarks from RISC-V benchmarks
- 288,789 gates in BOOM
- 114,750 gates in Rocket

Fast Power Analysis Results

- Error on power values:
  - Relative error < 1.5%
- Relative ranking on gates:
  - Kendall's τ > 0.97
  - > 0.9 with only 0.01% cycles

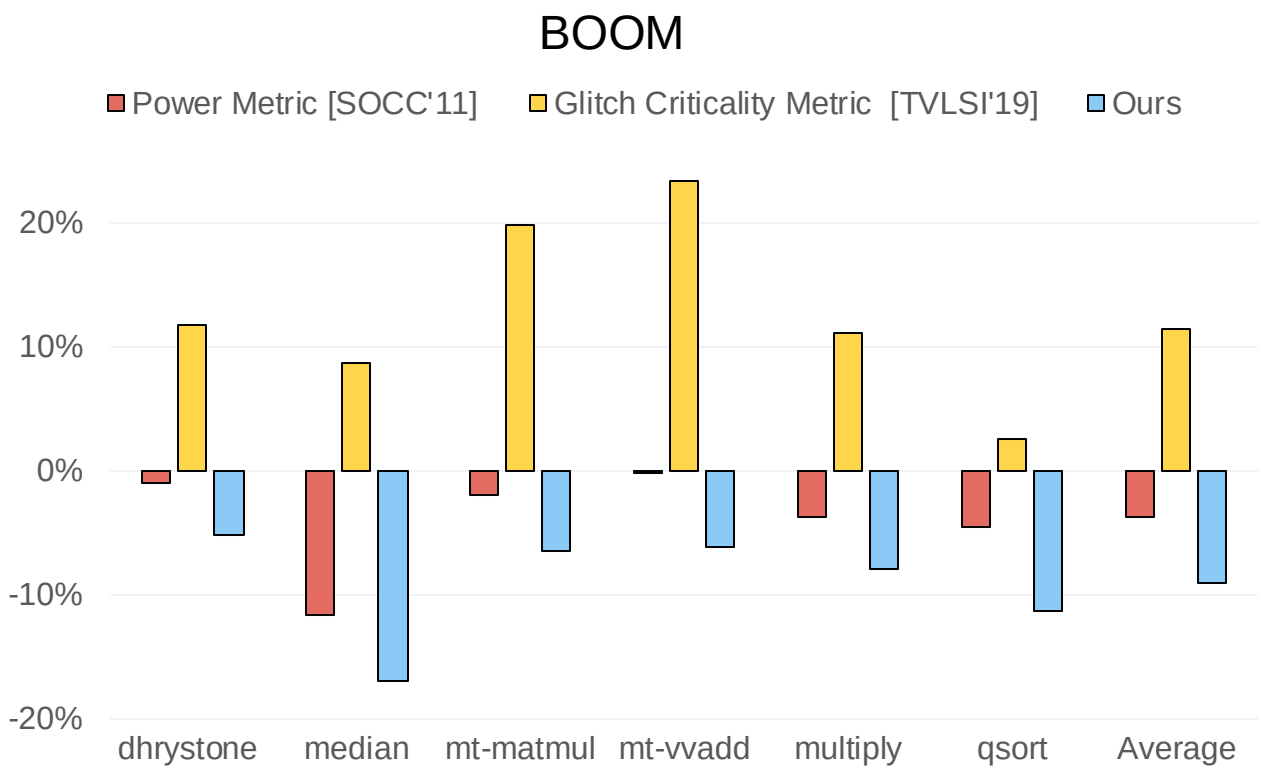


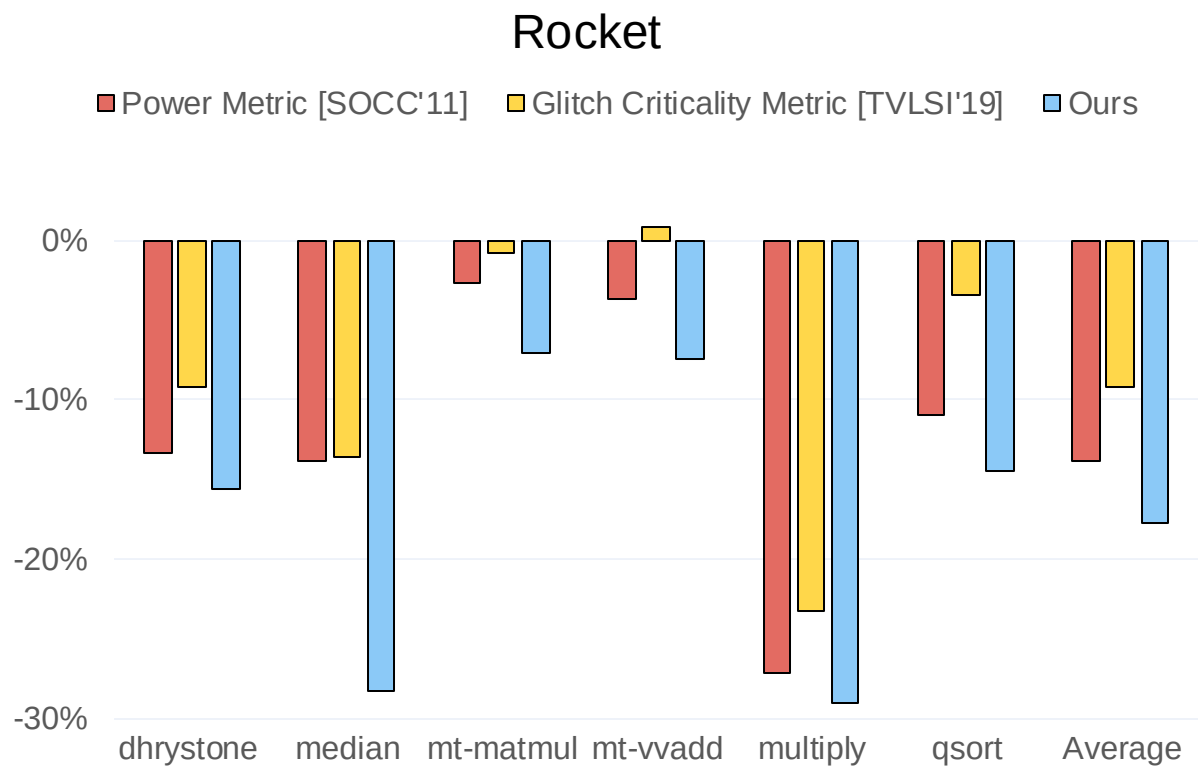




Glitch Power Optimization Results

- SMART-GPO
  - Always reduce glitch power
  - Largest glitch power reduction



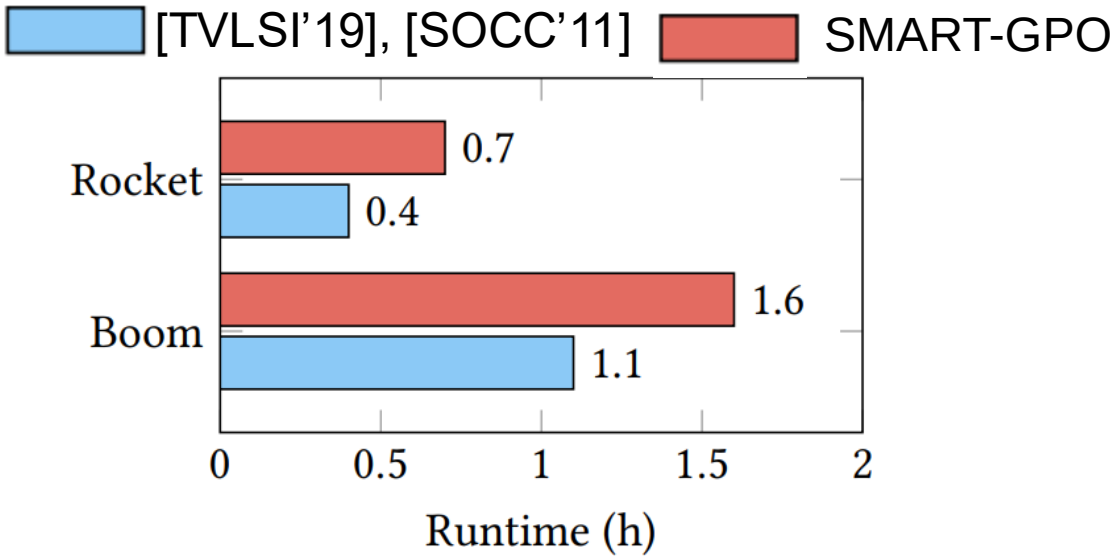


Timing & Area

- Timing:
  - Set frequency at 500MHz
  - No timing violation
- Area:
  - BOOM: 3590 gates optimized (↓1.2%)
  - Rocket: 1206 gates optimized (↓1.1%)
  - Marginal improvement
- Total power reduction:
  - 3% ~ 4%

Runtime

- Previous methods ([TVLSI'19], [SOCC'11])
  - Size gates one-by-one to avoid timing violation
- SMART-GPO
  - Additional sensitivity measurement step with fast estimation
  - Slightly slower



Remarks

- ✓A unified framework to analyze and optimize the glitch power based on sensitivity measurement.
- ✓Fast power analysis based on a small portion of sampled cycles.
- ✓Compute sensitivity based on sampling and statistical estimation.
- ✓A general methodology that can seamlessly accommodate all previous hand-crafted heuristics.

Thanks
&
Questions?

Backup: Sizing Details

- For baseline methods ([TVLSI'19], [SOCC'11])
  - Size 4000 gates on BOOM and 2000 on Rocket
  - These gates covers more than 80% total metric (glitch criticality metric, power metric, resp.) on BOOM and Rocket.
- SMART-GPO:
  - Size 3590 gates on BOOM and 1206 on Rocket.
  - Size less gates but reach higher glitch power reduction.
- Timing:
  - Similar to baselines, we query timing engine after sizing each gate to avoid timing violation.

Backup: Sensitivity Measurement

- SMART-GPO relies on sampling to measure sensitivity.
- More sampled batches → better optimization result.
  - In our experiment, we sampled 15 batches.
  - Even sampling 5 batches can surpass baselines.