2025 Design, Automation & Test in Europe Conference (DATE 2025)

CPP-SGS: Cycle-Accurate Power Prediction Framework via SNN and Genetic Signal Selection

Tong Liu, Zijun Jiang and Yangdi Lyu†
Microelectronics Thrust, The Hong Kong University of Science and Technology (Guangzhou)
†Corresponding author: yangdilyu@hkust-gz.edu.cn

***Abstract*—Effective power management is crucial for optimizing the performance and longevity of integrated circuits. Cycle-accurate power prediction can help power management during runtime. This paper introduces a Cycle-accurate Power Prediction framework via Spiking neural networks (SNNs) and Genetic signal Selection (CPP-SGS), which integrates SNNs and Genetic Algorithms (GAs) to predict real-time power consumption of chips. We apply GAs to select the most relevant signals as the input to SNNs to reduce the model size and inference time, making it well-suited for dynamic power estimation in real-time scenarios. The experimental results show that CCP-SGS outperforms the state-of-the-art approaches, with a normalized root mean squared error (NRMSE) of less than 1.6%.**

***Index Terms*—power consumption prediction, SNN, cycle-accurate**

I. Introduction

Accurate power prediction is essential for the optimization of design efficiency and sustainability in integrated circuits (ICs). Traditional power prediction methodologies primarily depend on commercial power analysis tools such as PrimeTime PX (PTPX). These tools require simulation waveforms as inputs to conduct power analysis and are extensively employed to ensure that the design fulfills all power requirements across every process corner.

In this paper, we introduce the Cycle-Accurate Power Prediction Framework via SNN and Genetic Signal Selection (CPP-SGS). This framework integrates SNNs with GAs for signal selection to enhance the precision of power prediction in ICs. Instead of manually selecting important input features for predicting power consumption, our method utilizes GAs to identify signals that significantly contribute to the accuracy of the predictive model. The choice of SNNs [1] as the machine learning component is motivated by their event-driven computation model. When supported by appropriate hardware designs, SNNs typically result in substantially lower power consumption compared to conventional DNNs.

II. Methodology

In this section, we introduce the Cycle-Accurate Power Prediction Framework via SNN and Genetic Signal Selection (CPP-SGS), which utilizes GAs to automatically select signals that are highly correlated with power dissipation and employs SNNs to train power models.

A. Power Estimation using SNNs

We apply an SNN model [1] to delineate the intricate relationships between toggle traces and power traces. This modeling technique harnesses the dynamic processing capabilities of SNNs, which are particularly adept at capturing temporal dependencies and non-linear interactions within data. Unlike traditional neural networks, SNNs integrate the concept of time directly into their architecture, making them ideally suited for applications where temporal dynamics are a key component of the data structure.

To train the SNNs, we utilize a dataset comprised of synchronized pairs of RTL signal values and corresponding power measurements. The training process involves adjusting the synaptic weights based on the timing of signal spikes, which represents the signal transitions, thereby aligning the model more closely with the actual power usage patterns observed in the PTPX outputs.

B. Progressive Genetic Algorithm for Signal Selection

With all signals included as the inputs of our SNN model, the per-cycle power can be accurately predicted. However, the huge number of inputs leads to an extremely large and complex model, making it costly to deploy on ICs. To address this problem, we introduce a progressive genetic algorithm (PGA) to select the signals with the largest impact on the accuracy, thus reducing the number of inputs and the model complexity. The algorithm procedure is demonstrated in Fig. 1, which is briefly divided into two parts: genetic signal selection and model re-training.

1) Genetic Signal Selection: Reducing the number of inputs while maintaining the high accuracy of the model can be interpreted as a typical single-objective optimization problem, and genetic algorithms (GA) [2] are widely used to solve this type of problem. In our case, the variables of the problem are the selections of the signals, and the objective is to maximize the model's accuracy.

2) Model Re-training: After T iterations of GA, the best selection mask in the population is selected for the model re-training. Using the selection mask, the unselected signals in the dataset are filtered out to create a masked dataset. A new SNN model with N_{target} inputs is re-trained with the masked dataset. The re-trained model will be much smaller than the initial large model with all signals, while its accuracy only degenerates slightly.

III. Experimental Results

To validate the effectiveness and efficiency of our framework, we conducted experiments using RocketChip [3], an

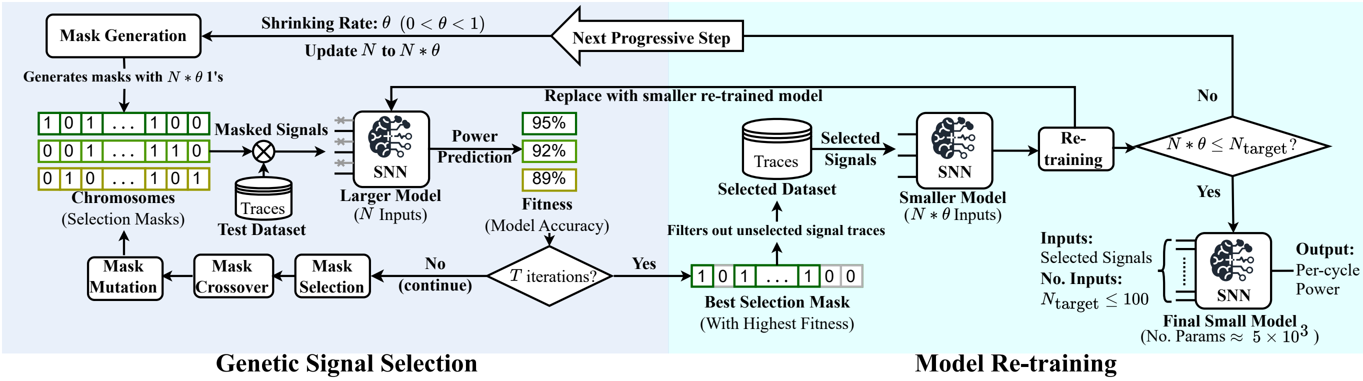

Fig. 1: Procedure of Progressive Genetic Algorithm (PGA)

open-source RISC-V processor, with all power consumption measurements computed using TSMC's 12nm technology. This approach enables a straightforward assessment of our power modeling techniques under practical conditions.

To demonstrate the effectiveness of our framework, we compared our predicted power consumption traces with those analyzed by PTPX from the test dataset. Additionally, we evaluated our results against PRIMAL [4], which employs different methods to predict power consumption in RocketChip.

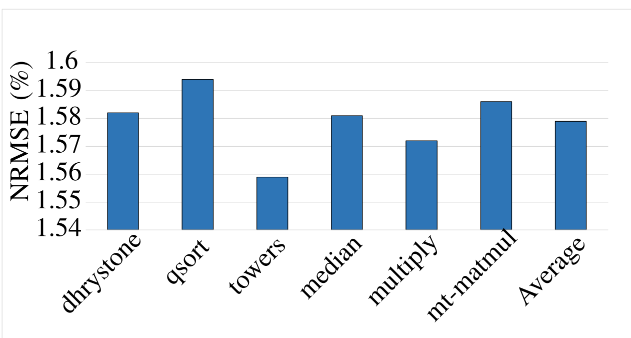

Fig. 2: Per-Cycle power Prediction Error Ratio

A. Per-Cycle Power Prediction Results

In our experiments, we employed Normalized Root Mean Square Error (NRMSE) as the evaluation metric for per-cycle power prediction. As depicted in Fig. 2, the CPP-SGS framework achieved a high accuracy rate. Across six different benchmarks, CPP-SGS maintained error rates below 1.6%. These results suggest that the input traces selected by the GAs effectively represent the power consumption signals, thereby validating the efficacy of the SNNs in our power prediction model.

Furthermore, we evaluate our work against PRIMAL [4], as depicted in Fig. 3. In the PRIMAL study, their CNN model utilizing default 2D encoding achieves an impressive 5.2% error rate on the test set. In comparison, their MLP, XGBoost, and Linear models report error rates of approximately 8%, 11%, and 13%, respectively. Notably, our framework demonstrates superior performance.

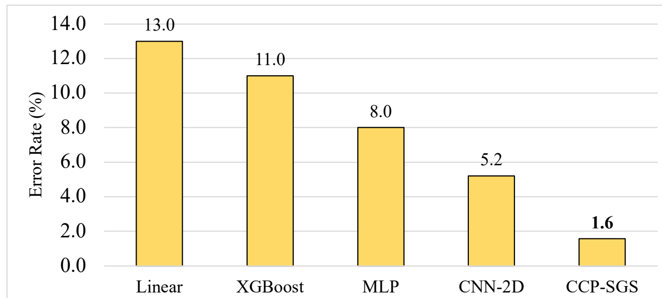

Fig. 3: Compare with other research works

B. Average Power Prediction Results

The average power consumption is readily computed from the per-cycle power trace, providing a straightforward method for assessing overall energy efficiency. In our study, we conducted a detailed comparison between the average power prediction errors derived from PTPX and those from our CPP-SGS framework. This comparative analysis reveals that our framework achieves a notably low error ratio, consistently less than 1%.

IV. Conclusion

In this work, we propose the CPP-SGS framework to effectively combine a GA for refined signal selection and an SNN for modeling to enhance power consumption predictions. The GA is employed to selectively identify critical signals, thereby optimizing the inputs for the SNN, which subsequently models complex, temporal power consumption patterns with high precision.

References

[1] X. Wang, X. Lin, and X. Dang, "Supervised learning in spiking neural networks: A review of algorithms and evaluations," *Neural Networks*, vol. 125, pp. 258–280, 2020.

[2] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[3] Asanovic *et al.*, "The rocket chip generator," UCB, Tech. Rep. UCB/EECS-2016-17, Apr 2016. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html

[4] Y. Zhou, H. Ren, Y. Zhang, B. Keller, B. Khailany, and Z. Zhang, "Primal: Power inference using machine learning," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.