IR drop Prediction Based on Machine Learning and Pattern Reduction

Yong-Fong Chang
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Yung-Chih Chen
National Taiwan University of Science and Technology
Taipei, Taiwan, R.O.C

Yu-Chen Cheng
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Shu-Hong Lin
National Taiwan University of Science and Technology
Taipei, Taiwan, R.O.C

Che-Hsu Lin
National Taiwan University of Science and Technology
Taipei, Taiwan, R.O.C

Chun-Yuan Chen
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Yu-Hsuan Chen
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Yu-Che Lee
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Jia-Wei Lin
Mediatek Inc
Hsinchu, Taiwan, R.O.C

Hsun-Wei Pao
Mediatek Inc
Hsinchu, Taiwan, R.O.C

Shih-Chieh Chang
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Yi-Ting Li
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

Chun-Yao Wang
National Tsing Hua University
Hsinchu, Taiwan, R.O.C

ABSTRACT

With the advances in semiconductor technology, the sizes of transistors are getting smaller, which has led to an increasingly severe impact of IR drop. Consequently, this trend has amplified the significance of IR drop analysis within the realm of chip design. However, analyzing IR drop is resource-intensive and time-consuming, since numerous simulation patterns are required to verify the power integrity of circuits. Additionally, with every engineering change order (ECO) step, a reevaluation is necessary. In this paper, we propose a machine learning-based method to predict IR drop levels and present an algorithm for reducing simulation patterns, which could reduce the time and computing resources required for IR drop analysis within the ECO flow. Experimental results show that our approach can reduce the number of patterns by approximately 50%, thereby decreasing the analysis time while maintaining accuracy.

This work is supported in part by the National Science and Technology Council (Taiwan) under MOST 109-2221-E-007-082-MY2, MOST 111-2221-E-007-121, MOST 111-2221-E-011-137-MY3, NSTC 112-2218-E-007-014, NSTC 112-2221-E-007-106-MY2, NSTC 112-2221-E-007-108, NSTC 112-2425-H-007-002, NSTC 113-2425-H-007-004, and National Tsing Hua University under NTHU 112A0241EX and NTHU 112A0316.

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658775

KEYWORDS

Dynamic IR drop analysis, IR drop prediction, pattern reduction

ACM Reference Format:
Yong-Fong Chang, Yung-Chih Chen, Yu-Chen Cheng, Shu-Hong Lin, Che-Hsu Lin, Chun-Yuan Chen, Yu-Hsuan Chen, Yu-Che Lee, Jia-Wei Lin, Hsun-Wei Pao, Shih-Chieh Chang, Yi-Ting Li, and Chun-Yao Wang. 2024. IR drop Prediction Based on Machine Learning and Pattern Reduction. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3649476.3658775

1 INTRODUCTION

With the advances in semiconductor technology, the transistor dimensions are getting smaller, which has brought some challenges in IC design, e.g., *IR drop* issue is a critical problem to be dealt with before design signoff.

IR drop refers to the voltage drop that occurs when current flows through the power delivery network (PDN). Unfortunately, IR drop harms the performance and reliability of the circuit, e.g., slower circuit operations due to a downgrade in the operating voltage and rise in chip's temperature due to increased resistance-generated heat. Hence, IR drop signoff has become an increasingly critical step in the design flow.

There are two approaches for analyzing a design's IR drop levels: dynamic analysis [9] and static analysis [5, 7]. Dynamic analysis simulates the circuit with patterns over time and analyzes the voltage drop that occurs under the different operations of the circuit. Hence, it is more computation-intensive and yields more precise estimations compared with static analysis.

Vector-based dynamic IR drop analysis usually requires significant time and computing resources due to numerous simulation patterns. A pattern refers to a sequence of input vectors used to simulate a circuit's behavior over a specified time frame. Analysis over a 30ns time frame may take up to 6.5 hours with commercial tools, e.g., Redhawk-SC [1] and Voltus [2]. Thus, having an efficient approach to IR drop analysis with accurate results is desired.

Several previous works [3, 4, 10, 11, 13–17] estimated IR drop based on machine learning (ML) techniques. Experimental results revealed that using ML models to predict IR drop is highly feasible.

In this paper, we also build an ML model to predict the IR drop. Based on accurate IR drop prediction results, we further introduce an algorithm aimed at minimizing the number of patterns necessitating simulation. In the current design flow, after completing an Engineering Change Order (ECO) task, designers need to reanalyze all the patterns. Our approach empowers ECO tasks to streamline the analysis by only focusing on the chosen patterns, which could reduce the time required for IR drop signoff.

2 BACKGROUND

2.1 Dynamic IR drop

Dynamic IR drop refers to the voltage drop in the PDN during transistor switching. When numerous cells switch concurrently at a high frequency, a higher current is pulled through the PDN, resulting in a fluctuating higher-voltage drop. Thus, dynamic IR drop significantly impacts the timing and reliability of circuits. Dynamic IR drop analysis focuses on the design's power management, timing optimization, and the effect of power supply noise. In this work, we consider dynamic IR drop.

2.2 ML model: eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) [12] is a machine learning model for supervised learning problems, e.g, classification and regression. It is an ensemble learning method that combines the results of multiple decision trees. XGBoost uses a gradient-boosting framework, which means that it iteratively builds models to reach the final model. It starts with a simple model and then adds more complex models to correct the errors made by the previous models. XGBoost provides a way to assess the importance of different features in making predictions, which is valuable for feature selection.

Compared to other types of models, XGBoost can build models and make predictions more quickly. Additionally, it works well for tabular data [8]. Thus, we use XGBoost as our ML model.

3 RELATED WORKS

In this section, we review some related works on IR drop prediction and address their issues for exploring potential improvements to enhance the model's accuracy.

Previous works [3, 4, 10, 11, 13–17] did not consider the package effect on IR drop prediction. In fact, the IC package is also an important factor affecting IR drop. After conducting the same preliminary experiments associated with the commercial tool, we observed a notable increase in IR drop when accounting for the package effect, as compared to when it was not taken into consideration. The main reason for this phenomenon would be the resistance and inductance of the package. Therefore, we incorporate this information into the features of our model in this work.

Although ML-based approaches significantly reduce the time spent on analyzing IR drop, errors exist in between the predicted IR drop and the golden IR drop. Therefore, designers could not entirely rely on the predicted values for IR drop signoff. However, they can leverage the predictions to identify critical *slice*s (which are smaller time frames divided from patterns) that are more likely to trigger instances of high IR drop for speeding up the IR drop analysis process [13]. Nonetheless, for front-end designers, each pattern has its specific importance, and selecting some slices arbitrarily from each pattern is meaningless for just reducing the time cost. Thus, we present an algorithm designed to choose a reduced number of patterns, significantly diminishing the time required for executing IR drop analysis in each ECO process.

4 PROPOSED APPROACH

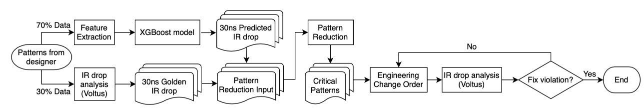


Figure 1: The overall flow of proposed approach.

The overall flow of the proposed approach is shown in Figure 1. When a design is completed, designers release patterns for design signoff. First, we divide the patterns into shorter slices and randomly select 30% of slices as the training data. We use commercial tools, e.g., Voltus, to compute the IR drop for the selected slices. Next, we train the proposed ML model, a two-level XGBoost-based model.

Then, we predict the IR drop for the remaining 70% of slices. After that, the golden IR drop of the 30% of slices and the predicted IR drop of the other 70% of slices are the input data of the pattern selection algorithm. Finally, we preserve patterns that cause high IR drop, called critical patterns, and discard the other patterns for the following ECO process.

4.1 Feature Extraction

The proposed model considers two categories of features, instance-based features and tile-based features, with 56 features in total.

Instance-based features: We use the automatic placement & routing (APR) tool to obtain the x-coordinate and y-coordinate of instances. Additionally, we use Voltus to get effective resistance (Reff) and resistance of least-resistance path (RLRP). Both Reff and RLRP have three features, which are the resistance at the power supply terminal, ground terminal, and both. We get bumps' resistance and inductance from the package information files. Like the Reff and RLRP, each bump resistance has three features since each instance has a bump at the power supply terminal and a bump at the ground terminal. Besides, we use the toggle rate and power information obtained from Voltus as our features.

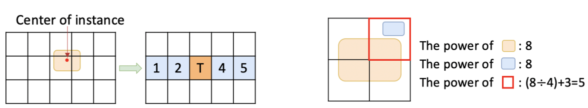
Toggle rate helps with timing information, measured as the average number of signal transitions per clock cycle. There are two types of toggle rate, toggle rate of the input (τ_i) and output pin (τ_o).

There are three basic types of power consumption in devices, i.e., internal power (P_i), switching power (P_s), and leakage power (P_l). Internal power refers to the power consumed by the active components of a circuit during operation. Switching power, also known as dynamic power, is consumed during the signal transition of circuits. Leakage power is consumed in a circuit in the standby or idle state. It comes from the small leakage currents in transistors. As transistor sizes continue to shrink, leakage power has significantly contributed to overall power consumption.

We also refer to the work [13] and add four power features: total power ($P_t = P_i + P_s + P_l$), scaled power ($P_{Scaled} = P_i \times \tau_i + P_s \times \tau_o + P_l$), overlapped-switching power, and overlapped-scaled power, into the proposed model. Overlapped power is obtained by summing up the power of the nine surrounding tiles of an instance. Using this, the model can be trained to learn the local effect of IR drop.

Tile-based features: Tile-based features are converted from instance-based features. The tile-based features are inspired by the neighbor cell feature in [15] and the density map feature in [6]. Powers (P_i, P_s, P_l, P_t, and P_{Scaled}) and toggle rates (τ_i and τ_o) are converted to tile-based features in addition to instance-based features. In this work, each instance's target tile is based on the center point of each instance like Figure 2(a).

Each instance includes information from five neighboring tiles as shown in Figure 2(a), allowing the model to learn the local effect of IR drop. For the conversion, we sum up the powers and toggle rates of all the instances in the tile. If there is an instance across multiple tiles, the value is calculated according to the cross-area ratio. Figure 2(b) is an example of conversion. We consider only five neighboring tiles in the same row since empirical experiments show they are enough to achieve a similar quality as [15] and [6].



(a) (b)

Figure 2: Instance-to-tile conversion method. (a)An instance's target tile. (b)Example of tile-based feature conversion.

In summary, we use 56 features for an input instance, including 21 instance-based and 35 tile-based features. Since we utilize the XGBoost model, which operates in a tabular manner for predictions, the output is the IR drop of the input instance.

4.2 XGBoost Model

We build a two-level XGBoost-based model. We first train an XGBoost model as a classifier to distinguish between high-IR and low-IR instances, where we define the threshold for high IR drop as 120mV, about 15% of the supply voltage. Subsequently, we train two XGBoost regressors to predict the IR drop for high-IR and low-IR instances, respectively. As shown in Figure 3, the testing data is first passed through the classifier to classify the instances into high-IR and low-IR instances. Then, the corresponding regressors are selected to predict the IR drop. The inputs of the three XGBoost models are the same, i.e., the 56 features.

4.3 Pattern Reduction

Among the simulation patterns, only a few of these patterns are IR-critical patterns, which can trigger high IR drop. The purpose of pattern reduction is to identify these IR-critical patterns.

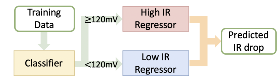


Figure 3: Inference phase of the proposed two-level model.

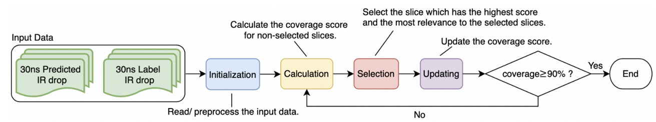


Figure 4: The flow of pattern reduction.

Figure 4 shows the flow of the algorithm. The labeled and the predicted IR drop are combined and then input to the pattern reduction algorithm. First, we calculate the critical instances coverage score for each non-selected slice, where a critical instance is the instance whose IR drop exceeds the threshold. The scoring algorithm unifies the critical instances of the slice with the selected slices. Please note that an instance may be identified as a critical instance for one slice but a non-critical instance for the other slices.

Next, we use a greedy approach to select the slices with the highest scores as candidates. Among these candidates, we choose the slice that has the most relevance to the selected slices, and add them to the selected slices. We also update the coverage score accordingly. We repeat the process until the coverage of critical instances reaches 90%.

Table 1 shows an example. There are three patterns (1^{st}, 2^{nd}, 3^{rd}) and they are divided into 6 slices, shown in *Slice* column, where the slice name starting with n means n^{th} pattern. The candidate count is set to 3. In the first iteration, the three candidates with the highest coverage scores are 1_*a*, 3_*a*, and 1_*b*. Thus, we choose 1_*a* in this iteration and the overall score is updated to 50. In the second iteration, based on the updated scores, the top three candidates are 3_*a*, 2_*a*, and 1_*b*. Although 3_*a* has the highest score, 1_*b* is prioritized since it belongs to the same pattern as the previously selected slice. Then, the overall score is updated to 70. In the third iteration, the highest scores are 3_*a*, 3_*b*, and 2_*a*. 3_*a*, which has the highest score, is chosen. After these iterations, the overall score reaches 95, surpassing the threshold of 90, and the algorithm is terminated. Finally, the selected slices are 1_*a*, 1_*b*, and 3_*a*. Thus, the chosen patterns are the 1^{st} pattern and the 3^{rd} pattern.

5 EXPERIMENTAL RESULTS

We conducted the experiments on an industrial design with 3,508,819 instances. There are nine 300ns simulation patterns, which are divided into 76 30ns slices. The supply voltage is 0.9V and the process technology is TSMC 5nm process. The length and width of a tile are twice the maximum instance length and width, respectively. The model is trained with the golden instance IR drop obtained from Voltus. To evaluate the quality, we used *Root Mean Square Error* (RMSE) and *Mean Absolute Error* (MAE).

5.1 IR Drop Prediction Results

The results are presented in Table 2. In this work, our primary objective is to decrease the number of simulation patterns rather

Table 1: An example for pattern reduction.

Slice	#Critical Instance	Iter. 1		Iter. 2		Iter. 3	
		sc	cand.	sc	cand.	sc	cand.
1_a	50	50	1	-	-	-	-
1_b	40	40	3	70	3	-	-
2_a	25	25		70	2	85	3
2_b	35	35		65		80	
3_a	45	45	2	75	1	95	1
3_b	35	35		65		90	2
total coverage score (0/100, initially)		50/100		70/100		95/100	
total chosen slices (empty, initially)		1_a		1_a, 1_b		1_a, 1_b, 3_a	
critical patterns						1, 3	

than achieving optimal IR drop prediction results. Nonetheless, it is essential to obtain reliable IR drop predictions to ensure the effectiveness of our proposed pattern reduction algorithm. Therefore, we present our prediction results to demonstrate the utility of the IR drop information fed into our algorithm, without making comparisons to previous works.

We used 30% of the slices for training and the other 70% for testing. To validate whether our model can predict the IR drop accurately, we attempted different combinations of training and testing data. For Set A, we randomly selected 23 slices out of the 76 slices as the training data. For Set B, we randomly selected 3 patterns out of the 9 patterns, and used the slices of the selected patterns as the training data. Finally, for Set C, we selected 23 slices having a better balance of high IR drop and low IR drop as the training data. We repeated the experiments for 5 times and recorded the average and the best results. For the best results, the MAE is 3.954mV and the RMSE is 5.462mV. The correlation coefficient is 0.95, which means that our predicted results are similar to the label.

The total CPU time is about 1 hour, about 1300 seconds for training, and about 2300 seconds for inference.

Table 2: The result of IR drop prediction by XGBoost.

Design 1		MAE	RMSE	MaxE	MinE
Set A	Normal	4.983	6.336	97.185	-139.944
	Best	4.970	6.265	88.279	-63.153
Set B	Normal	4.356	5.940	105.683	-145.045
	Best	4.344	5.854	93.665	-67.404
Set C	Normal	3.968	5.556	120.747	-139.504
	Best	3.954	5.462	93.486	-67.441

5.2 Pattern Reduction Results

To evaluate the effectiveness of pattern reduction, we establish the following criteria. We only select up to half of all patterns, and the total critical instance coverage score of the selected patterns must be greater than 90%.

Furthermore, for comparison, we use a brute-force algorithm to identify the golden critical patterns. By calculating the critical instance coverage score for each possible combination of patterns, we select the one with the highest score as the golden patterns.

We adjust the number of candidates (N) in the proposed algorithm for the experiments. When the threshold of critical IR drop is 120mV, the golden critical patterns are (1, 2, 3, 7) and (1, 2, 3) when we set the numbers of selected patterns to 4 and 3, respectively. Both combinations can achieve the total critical instance coverage score greater than 90%. The experimental results show that our algorithm can effectively identify critical patterns. When N is set to 1, 3, 5, and 7, our algorithm all chooses 11 slices and the same patterns as the golden critical patterns are chosen. When N is set to 1 or 3, patterns 1, 2, 3, 7 are chosen; when N is set to 5 or 7, patterns 1, 2, 3 are chosen.

For the CPU time of our pattern reduction algorithm, we need about 600 seconds; but for the brute-force algorithm, it takes 2 hours to identify the critical patterns. Thus, our algorithm can efficiently and effectively identify critical patterns for IR drop analysis.

6 CONCLUSION

We propose a two-level XGBoost-based model for IR drop prediction with the consideration of package effect, which improves the accuracy of IR drop prediction. Additionally, we propose an algorithm to identify critical patterns for saving the time and computing resources required for IR drop analysis within the ECO flow.

REFERENCES

[1] Ansys 2018. *RedHawk User Manual.* Ansys.
[2] Cadence [n. d.]. *Voltus User Manual.* Cadence.
[3] Chi-Hsien Pao et al. 2020. XGBIR: An XGBoost-based IR drop predictor for power delivery network. In *Design, Automation & Test in Europe Conference & Exhibition.* 1307–1310.
[4] Chia-Tung Ho et al. 2019. IncPIRD: Fast learning-based prediction of incremental IR drop. In *International Conference on Computer-Aided Design.* 1–8.
[5] Howard H. Chen et al. 1997. Power supply noise analysis methodology for deep-submicron VLSI chip design. In *Design Automation Conference.* 638–643.
[6] Jia-Xian Chen et al. 2022. Vector-based Dynamic IR-drop Prediction Using Machine Learning. In *Asia and South Pacific Design Automation Conference.* 202–207.
[7] Rishi Bhooshan et al. 2007. Optimum IR drop models for estimation of metal resource requirements for power distribution network. In *International Conference on Very Large Scale Integration.* 292–295.
[8] Ravid Shwartz-Ziv et al. 2022. Tabular data: Deep learning is not all you need. *Information Fusion* 81 (2022), 84–90.
[9] SK Nithin et al. 2010. Dynamic voltage (IR) drop analysis and design closure: Issues and challenges. In *International Symposium on Quality Electronic Design.* 611–617.
[10] Seyed Nima Mozaffari et al. 2019. An efficient supervised learning method to predict power supply noise during at-speed test. In *International Test Conference.* 1–10.
[11] Shih-Yao Lin et al. 2018. IR drop prediction of ECO-revised circuits using machine learning. In *VLSI Test Symposium.* 1–6.
[12] Tianqi Chen et al. 2016. Xgboost: A scalable tree boosting system. In *SIGKDD International Conference on Knowledge Discovery and Data Mining.* 785–794.
[13] Vidya A Chhabria et al. 2021. MAVIREC: ML-aided vectored IR-drop estimation and classification. In *Design, Automation & Test in Europe Conference & Exhibition.* 1825–1828.
[14] Vidya A Chhabria et al. 2021. Thermal and IR drop analysis using convolutional encoder-decoder networks. In *Asia and South Pacific Design Automation Conference.* 690–696.
[15] Yen-Chun Fang et al. 2018. Machine-learning-based Dynamic IR Drop Prediction for ECO. In *International Conference on Computer-Aided Design.* 1–7.
[16] Yonghwi Kwon et al. 2021. Dynamic ir drop prediction using image-to-image translation neural network. In *International Symposium on Circuits and Systems.* 1–5.
[17] Zhiyao Xie et al. 2020. PowerNet: Transferable dynamic IR drop estimation via maximum convolutional neural network. In *Asia and South Pacific Design Automation Conference.* 13–18.