IR Drop Prediction of ECO-Revised Circuits Using Machine Learning

Shih-Yao Lin[1], Yen-Chun Fang[1], Yu-Ching Li[1], Yu-Cheng Liu[1], Tsung-Shan Yang[1], Shang-Chien Lin[1],Chien-Mo Li[1]
Eric Jia-Wei Fang[2]
[1]Graduate Institute of Electronics Engineering
National Taiwan University, Taipei 106, Taiwan
[2]MediaTek Inc., Hsinchu 300, Taiwan

***Abstract* — Excessive *power supply noise* (PSN), such as IR drop, can cause timing violation in VLSI chips. However, simulation PSN takes a very long time, especially when multiple iterations are needed in IR drop signoff. In this work, we propose a machine learning technique to build an IR drop prediction model based on circuits before ECO (engineer change order) revision. After revision, we can re-use this model to predict the IR drop of the revised circuit. Because the previous circuit(s) and the revised circuit are very similar, the model can be applied with small error. We proposed seven feature extractions, which are simple and scalable for large designs. Our experiment results show that prediction accuracy (average error 3.7mV) and correlation (0.55) are very high for a three million-gate real design. The run time speedup is up to 30X. The proposed method is very useful for designers to save the simulation time when fixing the IR drop problem.**

Keywords — power supply noise, IR drop analyzer, machine learning

I. INTRODUCTION

Power supply noise (PSN) has become an important concern for VLSI system design and test [1, 2]. Excessive PSN degrades circuit performance, which even leads to timing failure [3, 4]. It is a well-known problem that excessive PSN can induce significant yield loss (overkill) [5, 6, 7]. PSN include IR drop and *Ldi/dt* noise. Since IR drop is more significant than the *Ldi/dt* noise for on-chip power integrity analysis, this paper will focus on the IR drop effect only.

Traditional dynamic IR drop analyzer solves large linear equation systems to obtain the IR drop of every node in the circuit, and then simulate critical paths to verify if there is any IR drop violation [8]. However, this process is very slow, especially when multiple iterations are needed in IR drop signoff. For an industry scale design (~3M gate count), IR drop analysis can take up to one day. Every time a minor revision is made, the whole process has to be repeated, even if the revised circuit just changed a small number of cells.

It has been shown that machine learning prediction of circuit speedpath [9] and timing signoff [10] is feasible. Recently, Ye et al.[11] developed an SVM-based regression method to predict circuit delay at runtime without PSN consideration. However, it has been shown that IR drop analysis is inaccurate if PSN is ignored [12]. Unfortunately, realistic large circuits are difficult for machine learning since the dimension is very large. Power-aware dynamic IR drop prediction of cells can be found in [13]. They used linear model to predict the IR drop of cells. However, the prediction rule is based on designer's experience, which cannot be generalized and automated. So far, there is still no good machine learning technique available to predict PSN for large circuits.

Fig. 1 shows the traditional flow of IR drop analysis. After each circuit revision, we need to rerun the IR drop analyzer to make sure there is no violation. The source of patterns can be either functional patterns or test patterns. Because real design process needs many revisions, repeated IR-drop analysis during each iteration can be very time consuming.

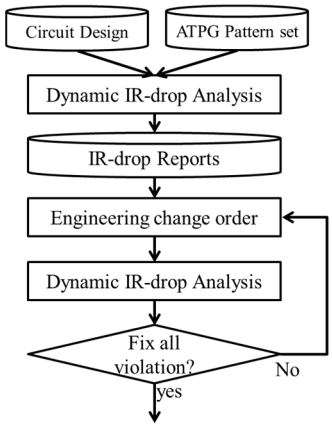


Fig. 1. Traditional IR drop analysis flow

In this work, we propose to use machine learning to build an IR drop prediction model for the circuit(s) before revision. After a circuit revision, we can re-use this model to predict the IR drop of the revised circuit. After the predicted IR drop meets our specification, we need to rerun the dynamic IR drop analyzer again to make sure there is indeed no violation before the final signoff. This work has three major contributions. We take advantage of the similarity between the original circuit and the revised circuit to learn a model to speed up the signoff process so very few dynamic IR drop analyses are needed. This new

978-1-5386-3774-6/18/$31.00 ©2018 IEEE

flow saved a lot of iterative simulation time during revision. Second, we propose to sample a small portion of cells to predict IR drop of all cells. This greatly reduces the size of input data so that machine learning of realistic industrial design is feasible. Third, we propose seven simple but important feature extraction methods to greatly reduce the dimension, so the proposal is scalable for large designs. Our experiment results on a three million-gate GPU show that average error of prediction IR drop is 3.7mV and correlation is 0.55. The run time speedup is up to 30X compared to a commercial tool *Ansys RedHawk*. The proposed method is very useful for designers to save the simulation time during ECO to fix IR drop problems.

The rest of this paper is organized as follows. Section II provides previous research papers in PSN-aware IR analysis. Section III presents the proposed machine learning technique. Section IV shows experimental results on benchmark circuits. Finally, Section V concludes this paper.

II. Past Research

A. Statistical IR drop Prediction

Many different metrics have been proposed as alternatives to IR drop, such as *weighted switching activity* (WSA) [14, 15], *switching cycle average power* (SCAP) [16], *flip-flop toggle count* (FFTC) [17], and *etc*. Although some metrics show good correlations with actual IR drop values, there is no known model to translate the proposed metrics to the actual IR drop values. It is not clear what is the pass/fail threshold for these metrics. Therefore, it is impossible to use these alternative metrics to sign off a design. A recent paper used a linear model to predict the IR drop values [13]. For each cell, they calculated a linear model to predict the IR drop based on the power consumption. The problem of the linear model is that it may not be good enough for complex designs. In addition, it is computationally expensive to calibrate a linear model for each cell in large designs. A paper tried to identify high power area (*hot-spot*) using switching probability and logic level [18]. Although we see a correlation between real hot-spot and the predicted area, it is still not clear what is the pass/fail threshold for design sign off.

B. Machine Learning IR drop Prediction

Machine learning has been applied to identify speedpath outliers [9]. Various feature extractions have been performed based on topology, dynamic effects, static effects, statistical effects, and random effects. Nevertheless, it did not consider IR drop effects. Support vector machine has been applied to predict IR drop [11]. This technique was implemented on FPGA to dynamically adjust the CPU operation frequency. Their technique used only input patterns, no feature extraction, to predict IR drop. The number of dimensions is very large and therefore it is not scalable for large designs. Another previous work is IR-drop-aware timing prediction using machine learning [19]. This work proposed feature extraction so it is scalable for large designs. However, it did not consider the ECO revision issue. Every time a new revision is made, a new model is needed.

C. Dynamic IR drop Analyzer

Our proposed machine learning technique can be applied to speed up any circuit IR drop analyzer. In this paper, we use a PSN-aware dynamic IR drop analyzer, *IDEA* (*IR drop-aware Efficient timing Analyzer*) as our benchmark simulator [12]. This technique is very scalable because they model the voltage-delay characteristic function in a simple analytical function, which just require limited simulation of library cells. Experimental results showed that, for small circuits, the error is less than 5% compared with HSPICE. Although IDEA is up to 272 times faster than a commercial tool, NANOSIM, it still takes days to simulate million-gate designs.

III. Proposed Technique

A. Proposed Flow

Fig. 2 shows the proposed flow of our work. During the design phase, we have several ECO-revised circuits, including *previous versions* (…, E_{n-2}, E_{n-1}) and *the current version* (E_n). Suppose that we have performed dynamic IR drop analysis on previous versions, using dynamic IR drop analyzer, such as *Ansys Redhawk*[8]. We can then extract important features from a small number of *sampled cells*. After that, we run a machine learning to build a *model* for this circuit so we can re-use this model to predict IR drop of the current version (E_n). Designers can use our prediction results to quickly evaluate whether IR drop of the current version meets the specification or not. Our machine learning prediction can save a lot of IR drop analysis runtime during iterations. Finally, when the predicted IR drop all meet our specification, we need to run the dynamic IR drop analyzer again to make sure there is indeed no violation before the final signoff. Compare Fig. 2 with Fig. 1, we can save simulation time during the prediction phase.

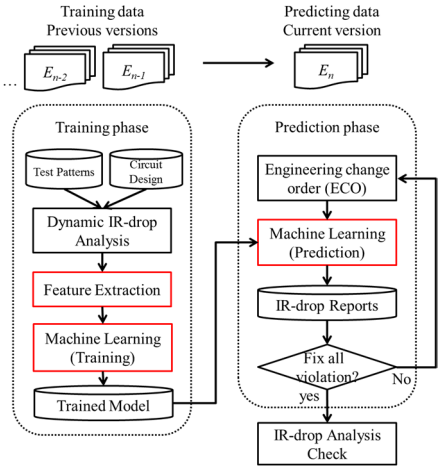


Fig. 2. Proposed IR Drop Prediction flow

B. Cell Sampling and Feature Extraction

Because there are many cells in a real design, it is impractical to use all cells to build a machine learning model. In this research, we propose to sample a portion of cells to build a model. Two factors should be considered when we take samples: (1) physical location and (2) IR drop values. For (1), we divide the chip layout into *M* x *N* windows. Based on our experience, we take 5% ~ 10% sampling cells randomly from each window. For (2), we sort all cells by their IR drops. We take sampled from three categories of cells: serious IR drop, medium IR drop cells, and low IR drop.

Table I shows the features we consider in this work. Given a sampled cell, there are three categories of features. *Power features* of a sampled cell include the power of this cell, toggle rate of this cell, and type of this cell. *Physical features* include this cell location (*i.e. X, Y* coordination), toggle rate of neighbor cells, and neighbor count (number of cells in the neighborhood). Finally, the *via feature* is the distance to via. Each feature is explained as follows.

TABLE I. SEVEN FEATURES OF A SAMPLED CELL

categories	1	2	3
Power features	Cell power	Cell toggle rate	Cell type
Physical features	Cell location	Neighbor toggle rate	Neighbor count
Via feature	Distance to via		

Cell power is the power consumption of the sampled cell given a set of input patterns. *Cell toggle rate* measures the switching activity of the sampled cell. *Toggle rate* is defined as the number of toggles over the number of clock cycles and it is a number between 0% and 200%. The reason for 200% toggle rate is because clock buffers toggle twice in each cycle. Both cell power and cell toggle rate are scalar variables that can be obtained by a dynamic IR drop analyzer, such as *Redhawk*. *Cell type* is the logic gate type of the sampled cell, such as NAND, NOR, and *etc*. This is a categorical scalar variable, which can be obtained from the netlist or the IR drop analysis report.

Fig. 3 shows how to define neighbors for a given sampled cell. We draw a rectangle window, centered at the given sampled cell. The window height and width can be adjusted by the user. Different technology may have different setting. In this work, we increasingly enlarge the window size and observed the prediction accuracy under different window width and height. After several experiments, our ANN model reached the highest prediction accuracy when window height is set to three row heights and window width is 50.

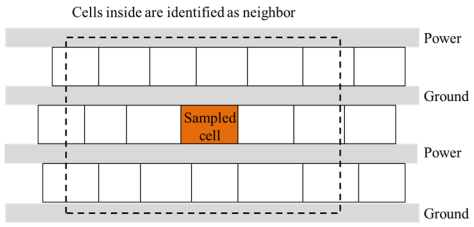


Fig. 3. Neighbors of a sampled cell

Neighbor toggle rate (NTR) is the toggle rate among all neighbor cells. Because different cell types have different impact on IR drop, so we need to count *NTR* according to cell types. For each cell type, toggle rates of the same cell type are summed up. This feature is a vector, whose dimension equals to the number of cell types. *NTR* of a sampled cell *s* is shown as the following equation (1).

$$NTR(s) = [\sum_{k\in W} TR_k^{type=t_1}, \ \dots\ , \sum_{k\in W} TR_k^{type=t_N}] \quad (1)$$

, where TR_k^t indicates the toggle rate of the k_{th} cell of type *t*, and *W* is the neighbor window of the sample cell *s*.

Neighbor count (NC) means the total count of neighbor cells. This feature is a scalar, defined in the following equation (2), where $Cell_k$ is an indicator variable (1 means the presence of the k_{th} cell, and 0 otherwise).

$$NC(s) = \sum_{k\in W} Cell_k \quad (2)$$

Distance to via (D) is the distance to the closest power via, and this via must be in the same row as the sampled cell. Fig. 4 shows the definition of *D*. The sampled cell is in the middle of the window and the red rectangles represent power vias. Number *D* represents the resistance value from the sampled cell to the power network.

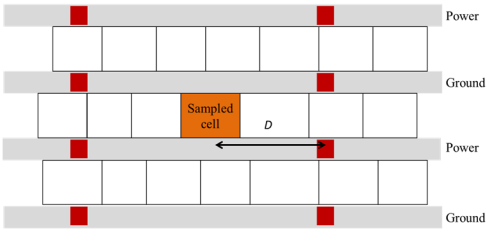


Fig. 4 Distance to via

Totally, we propose seven features of dimension (*T*+7), where *T* is the total number of cell types used in the design. This is a very small dimension and scalable for large designs.

C. Machine Learning Prediction Model

Artificial neural network (ANN) [20] imitated the neural structure of human's brain. Figure 5 shows an example ANN model with one hidden layer, where x_n is the input features of n_{th} sampled cell, and t_n is the target IR drop value of the n_{th} sampled cell. *w* is the weight of neurons in ANN and *N* is the number of training data.

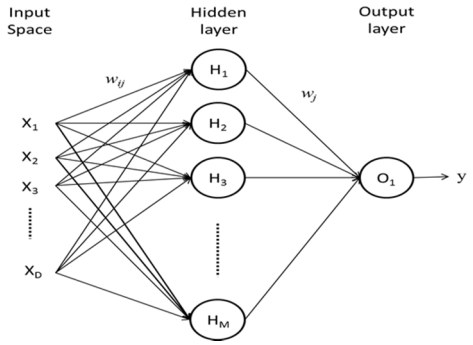


Fig. 5 ANN Model (with one hidden layer)

Our goal is to find a function $y(x_n, w)$ to minimize the following error function,

$$E(w) = \frac{1}{2}\sum_{i=1}^{N}\|y(x_n, w) - t_n\|^2 \quad (3)$$

IV. Experimental Results

Three ITC'99/IWLS'05 benchmark circuits (*b18, b19, leon3mp*) in 45nm and one real GPU (*Graphic Processor Unit*) in 16nm technology from the industry have been evaluated by our proposed method. Profiles of these four circuits are shown in Table II. The first column shows the number of cells. Given a commercial ATPG test pattern set, three ITC/IWLS benchmark circuits have been simulated by our own dynamic IR drop simulator, IDEA [12]. The second column shows the number of patterns simulated. The max IR drop and the average IR drop of the circuit are shown in the third and fourth columns. Dynamic IR drop analysis of the GPU was performed by a commercial tool, *Ansys RedHawk*. All the machine learning experiments use the artificial neural network open source FANN [21]. The experiments were run on Intel Xeon CPU E5520 @ 2.27GH with 32GB RAM.

TABLE II. Profile of Benchmark Circuits (E_1 before ECO)

Circuit	Cells	Patterns	V_{DD} (V)	Avg. IR drop(mV)	Max IR drop(mV)
b18	64K	50	1.1	29	39
b19	128K	50	1.1	59	83
leon3mp	638K	50	1.1	92	241
GPU	3,006K	240	0.9	25	190

A. IR Drop Prediction before ECO

We first evaluate the effectiveness of the IR drop prediction for the circuit before engineering change order (ECO). Prediction accuracy is measured by *Normalized root mean square error (NRMSE),* which is defined in equation (4) and (5). In these equations, $\hat{y}_i$ is the simulated IR drop of the i_{th} sample cell, and y_i is the predicted IR drop of the i_{th} sample cell. N is the number of data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \quad (4)$$

$$NRMSE = \frac{RMSE}{mean(y)} * 100\% \quad (5)$$

First, we want to know how many samples we need to build a model with high prediction accuracy. In this experiment, we sampled a small portion of cells and predict IR drop of all cells. The training data and predicting data are from the same design (E_1). There is no ECO-revision in this experiment. The prediction accuracy for three benchmark circuits is plotted in Fig. *6*. We observe that *NRMSE* drops quickly when sampling cells increase. We can see that *NRMSE* remain constant when the percentage of sampled cells is more than 10%. These experiments show that 10% sampling is enough for our designs. Please note that IR drop is highly design-dependent. Each design has a unique model, even if they are the same technology.

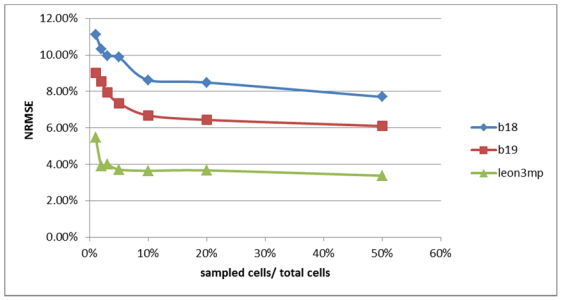


Fig. 6. Prediction accuracy vs. number of samples (before ECO)

Table III displays prediction results of four benchmark circuits. The machine learning model is trained by data from 10% sampled cells of the first edition E_1. Both *NRMSE* and *CC* are very good. As shown in the table, machine learning can predict IR drop accurately, without any ECO-revision, compared to simulation results. To evaluate the ANN technique, we also tried the *extra tree* technique [22]. Results of three benchmark circuits are very similar to those of ANN.

TABLE III. Experiment Results (Training=Prediction=E_1)

Circuit	Feature Dimension	NRMSE ANN, Tree	CC ANN, Tree
b18	42	8.7%, 6.8%	0.94, 0.95
b19	44	6.6%, 6.1%	0.94, 0.94
leon3mp	55	3.3%, 4.4%	0.98, 0.98
GPU	1,201	6.7%, NA	0.78, NA

Correlation coefficient (CC) is defined in equation (6). Smaller *NRMSE* and bigger *CC* indicates better results.

$$CC = \frac{\sum_{i=1}^{N}[y_i - mean(y)]\,[\hat{y}_i - mean(\hat{y})]}{\sqrt{\sum_{i=1}^{N}[y_i - mean(y)]^2 \;\sum_{i=1}^{N}[\hat{y}_i - mean(\hat{y})]^2}} \quad (6)$$

B. IR Drop Prediction after ECO

We evaluate the effectiveness of the IR drop prediction for circuits after ECO. First, we use the original circuit as edition E_1. Then, we use *Cadence SOC Encounter* to move 13 and 7 serious IR drop cells in benchmark circuits b18 and b19, respectively, to produce a new edition E_2. For benchmark circuit leon3mp, we add one power stripe to produce edition E_2. Then we move 32 cells to produce

edition E_3. Three editions of GPU are real data from MediaTek. Table IV shows the prediction accuracy of four circuits after ECO. Machine learning model is trained by data from 10% sampled cells of the first edition (E_1). And then we use the model to predict the second (E_2) and the third edition circuits (E_3).

TABLE IV. PREDICTION RESULTS OF FOUR CIRCUITS AFTER ECO

	E_1		E_2		E_3	
Circuit	NRMSE	CC	NRMSE	CC	NRMSE	CC
b18	8.7%	0.94	11.2%	0.88	-	-
b19	6.6%	0.94	9.7%	0.93	-	-
leon3mp	3.3%	0.98	6.1%	0.98	7.7%	0.98
GPU	6.7%	0.78	9.0%	0.59	11.2%	0.61

We can see from Table IV that our machine learning model has the best prediction accuracy when predicting the first edition circuit. E_1. As the number of revision increases, prediction accuracy becomes worse. Therefore, it is important to train the model using the most recent revision. Table V and Table VI use both previous editions (E_1 and E_2) data to improve the prediction accuracy of the third edition (E_3). Table V shows the prediction results of randomly sampled 10% cells in E_3. Table VI shows the prediction results of top 10% serious IR drop cells in E_3. With both E_1 and E_2 data in the training, the prediction accuracy is much better than that of using E_1 data only (Table IV). Average error is defined in equation (7). Max Error is defined in equation (8).

$$\text{Average Error} = \frac{\sum_{i=1}^{N}\|\hat{y}_i - y_i\|}{N} \quad (7)$$

$$\text{Max Error} = \max\left(\hat{y}_i - y_i\right),\ i = 1 \text{ to } N \quad (8)$$

where $\hat{y}_i$ and y_i are simulated IR drop and predicted IR drop of the i_{th} sample cell, respectively. A positive error means under-prediction but a negative error means over-prediction. The average error of leon3mp is 5mV, 5% of the average IR drop values. The average error of GPU is 3.7mV, which is 15% of the average IR drop values. The max error is 40mv, which is about 20% of the worst case IR.

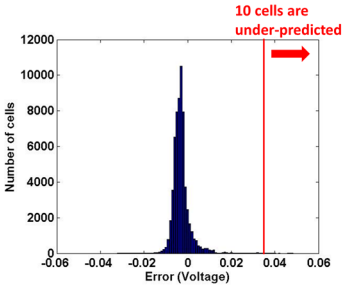
TABLE V. PREDICTION RESULTS OF E_3 CIRCUIT (TRAINED BY E_1+E_2)

	E_3		
Circuit	NRMSE	CC	Avg. Error
leon3mp	3.4%	0.98	3.8mV
GPU	6.8%	0.81	3.3mV

TABLE VI. PREDICTION OF TOP 10% SERIOUS IR DROP CELLS OF E_3

	E_3					
Circuit	NR MSE	CC	Avg. IR drop (mV)	Avg. Error (mV)	Max IR drop (mV)	Max Error (mV)
leon3mp	3.7%	0.54	92	5.0 (5%)	241	49.0 (20%)
GPU	7.4%	0.55	25	3.7 (15%)	190	39.3 (21%)

Fig. 7 is error distribution of leon3mp and GPU (top 10% worst cells in Table VI). Totally 60K and 300K cells for leon3mp and GPU, respectively. 99.9% of errors are smaller than 15% of max IR drop (36mV to leon3mp and 28.5mV to GPU). Red lines mean the 15% boundary. Only 10 cells (out of 60K) in leon3mp and 22 cells (out of 300K) in GPU are under-predicted.



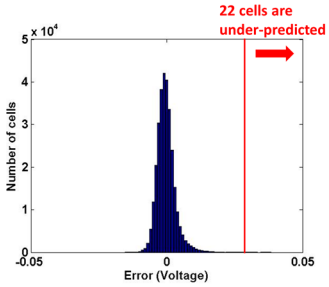
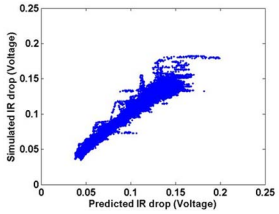


Fig. 7a. Leon3mp error distribution (60K cells)

Fig. 7b. GPU error distribution (300K cells)

Fig. 8 shows the plot of simulation IR drop results versus predicted IR drop for leon3mp and b18. Training data are E_1 plus E_2 and prediction data is E_3. Y axis represents simulated IR drop. X axis represents predicted IR drop. Correlation of simulated IR drop and predicted IR drop is 0.98 for leo3map and 0.88 for b18.



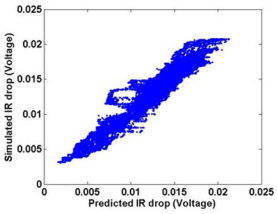


Fig. 8(a). Leon3mp **Fig. 8(b). b18**

Fig. 9 shows the IR drop map of leon3mp E_3 circuit. Fig. 9a is simulated IR drop map and Fig 9b is predicted IR drop map. Green area is low IR drop area, yellow area is medium IR drop area, and orange area is serious IR drop area. Red dots are high IR drop cells. Correlation between simulated IR drop map and predicted IR drop map is high.

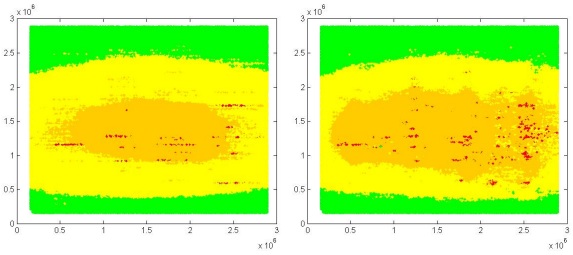


Fig. 9a. Leon3mp simulated IR drop map

Fig. 9b. Leon3mp predicted IR drop map

C. Runtime

Table VII shows the runtime comparison between proposed technique and commercial tools. In the proposed flow, we only need one feature extraction plus one training in the training phase. In the prediction phase, we need one feature extraction plus one (or more) prediction. Total time of proposed technique is two feature extraction time plus one training time plus one prediction time. Although we cannot save time for small circuits, we can save a significant amount of simulation time for large circuits. We need only 13 minutes to predict IR drop for GPU whereas *RedHawk* needs almost one day to simulate the circuit. The run time speedup is shown in the parenthesis (including feature extraction and training). Our technique significantly reduces the IR drop simulation time.

TABLE VII. RUNTIME COMPARISON

Circuit	b18	b19	leon3mp	GPU
Feature Extraction	12s	32s	147s	11m27s
Training	51s	106s	204s	24m57s
Prediction	1s	2s	19s	1m29s
Total time	76s	172s	517s	49m20s
NANOSIM	46s (0.6X)	95s (0.55X)	734s (1.4X)	-
RedHawk	-	-	-	1 day (30X)

V. DISCUSSION

For ANN to work well, both the number of hidden layers and neurons should be carefully tuned. Using too few neurons will result in *underfitting*. It occurs when there are too few neurons to detect important information in a large data set. Too many neurons may lead to *overfitting*. In this work, we tried two, three up to four hidden layers. We found that two hidden layers with twenty neurons for each hidden layer are enough for our data set. Too many layers would not improve the accuracy, and too many neurons would lead to overfitting.

Our proposal is good for the design sign-off stage, when the revised circuit is very similar to its previous version. Every time we add a new version, we would need to add this new version to our training so that this assumption can be valid.

VI. CONCLUSIONS

In this work, we have proposed an IR drop prediction for ECO-revised circuits using artificial neural network. We sampled a small portion of cells on a die to train the neural network. We proposed seven feature extractions, which are simple and scalable for large designs. Our experiment results show that prediction accuracy (average error 3.7mV) and correlation (0.55) are very high for a 3 million-gate real design. The run time speedup is up to 30X. The proposed method is very useful for designers to save the simulation time to fix the IR drop problem.

REFERENCES

[1] Kenneth L. Shepard, and Vinod Narayanan. "Noise in deep submicron digital design." *Proc. international conference on Computer-aided design*. IEEE Computer Society, 1997, pp. 524-531.

[2] Mohammad Tehranipoor, and Kenneth M. Butler. "Power supply noise: A survey on effects and research." *IEEE Design & Test of Computers* 2.27 (2010): 51-67.

[3] Howard H. Chen, and David D. Ling. "Power supply noise analysis methodology for deep-submicron VLSI chip design." *Proc. Design Automation Conference*. ACM, 1997, pp. 638-648.

[4] Yi-Min Jiang, and Kwang-Ting Cheng. "Analysis of performance impact caused by power supply noise in deep submicron devices." *Proc. Design Automation Conference*. ACM, 1999, pp. 760-765.

[5] L.-C. Wang, D.M.H. Walker, A. Majhi, B. Kruseman, G. Gronthoud, L.E. Villagra, P. van de Wiel,S. Eichenberger, "Power supply noise in delay testing," *Proc. International Test Conference,* pp. 1-10, 2006.

[6] Yi-Hua Li, et al. "Capture-power-safe test pattern determination for at-speed scan-based testing." *IEEE Transactions on computer-aided design of integrated circuits and systems* 33.1 (2014): 127-138.

[7] P. Girard, CW Wu, X Wen, *Power-aware testing and test strategies for low power devices,* 2010.

[8] *Apache RedHawk User Manual*, 2015.

[9] P. Bastani, K. Killpack, Li.-C. Wang, E. Chiprout, "Speedpath prediction based on learning from a small set of examples," *Proc. Design Automation Conference*, 2008, pp. 217 - 222.

[10] Andrew B. Kahng, Mulong Luo, and Siddhartha Nath. "SI for free: machine learning of interconnect coupling delay and transition effects." *System Level Interconnect Prediction (SLIP), 2015 ACM/IEEE International Workshop on*. IEEE, 2015, pp. 1-8.

[11] Fangming Ye, Firouzi, F., Yang Yang, K. Chakrabarty, M.B. Tahoori, "On-chip voltage-droop prediction using support-vector machines," *Proc. VLSI Test Symposium*, 2014.

[12] C.-Y. Han, Y.-C. Li, H.-T. Kan, and James C.-M. Li, "Power-Supply-Noise-Aware Test Pattern Analysis and Regeneration for Yield Improvement," IEICE, Vol.E99-A,No.12,pp.-,Dec. 2016.

[13] Yuta Yamato, "A Fast and Accurate Per-Cell Dynamic IR drop Estimation Method for At-Speed Scan Test Pattern Validation," *Proc. International Test Conference* 2012.

[14] Jeremy Lee, et al. "Layout-aware, IR-drop tolerant transition fault pattern generation." *Proc. Design, automation and test in Europe*. ACM, 2008, pp. 1172-1177.

[15] Junxia Ma, Jeremy Lee, and Mohammad Tehranipoor. "Layout-aware pattern generation for maximizing supply noise effects on critical paths." *2009 27th IEEE VLSI Test Symposium*. IEEE, 2009, pp. 221-226.

[16] N. Ahmed and M. Tehranipoor, "Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design," *Proc. Design Automation Conference*, 2007, pp. 533-538.

[17] Xiaoqing Wen, et al. "Low-capture-power test generation for scan-based at-speed testing." *IEEE International Conference on Test,* IEEE, 2005, pp. -1028.

[18] Kohei Miyase, et al. "Identification of high power consuming areas with gate type and logic level information." *2015 20th IEEE European Test Symposium (ETS)*. IEEE, 2015, pp. 1-6.

[19] Yu-Cheng Liu, Cheng-Yu Han, Shih-Yao Lin, and James Chien-Mo Li, "PSN-aware Circuit Test Timing Prediction using Machine Learning," *Proc. IET Computers & Digital Techniques*, 2016.

[20] Warren S. McCulloch, and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5.4 (1943): 115-133.

[21] Fast Artificial Neural Network. Available: http://libfann.github.io/fann/docs/files/fann-h.html

[22] P. Geurts, and *et. al.* "Extremely Randomized Trees," Machine Learning, Springer, 2006.