Generative Adversarial Network Based Scalable On-chip Noise Sensor Placement

Jinglan Liu*, Yukun Ding*, Jianlei Yang†, Ulf Schlichtmann‡, Yiyu Shi*
*Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556
Email:{jliu16, yding5, yshi4}@nd.edu
†Fert Beijing Research Institute,
Beijing Advanced Innovation Center for Big Data and Brain Computing,
School of Computer Science and Engineering, Beihang University, Beijing, China
Email: jianlei@buaa.edu.cn
‡Technical University of Munich, Munich, Germany
Email:ulf.schlichtmann@tum.de

***Abstract*—The relentless efforts towards power reduction of integrated circuits have led to the prevalence of near-threshold computing paradigms. With the significantly reduced noise margin, therefore, it is no longer possible to fully assure power integrity at design time. As a result, designers seek to contain noise violations, commonly known as voltage emergencies, through various runtime techniques. All these techniques require accurate capture of voltage emergencies through noise sensors. Although existing approaches have explored the optimal placement of noise sensors, they all exploited the statistical modeling of noise, which requires a large number of samples in a high-dimensional space. For large scale power grids, these techniques may not work due to the very long simulation time required to get the samples. In this paper, we explore a novel approach based on generative adversarial network (GAN), which only requires a small number of samples to train. Experimental results show that compared with a simple heuristic which takes in the same number of samples, our approach can reduce the miss rate of voltage emergency detection by up to 65.3% on an industrial design.**

I. Introduction

To meet the stringent power budget imposed by either package cooling limits or battery-life requirements in mobile applications, near-threshold computing has been explored in a broad range of applications, where the supply voltage is approximately equal to the threshold voltage of the transistors [1]. However, near-threshold computing significantly reduces the noise margin, making design-time power integrity assurance a very challenging task. There are a few more papers [2], [3] utilizing on-chip noise sensors to track voltage emergencies.

In order to accurately track voltage emergencies, it is critical to place the noise sensors optimally. Towards this, [4]–[7] proposed Eagle-Eye, the first noise sensor placement work in the literature to minimize the miss rate of voltage emergency detection [7]. The method is based on statistical modeling of noise and uses a heuristic which is proved to be optimal among all polynomial complexity algorithms. Later [8] further developed an alternative noise sensor placement framework but with the target of recovering the entire noise map [1] given that some of the functional areas cannot allow sensor placement. It is not quite relevant to the detection of voltage emergencies.

A fundamental issue with all these approaches is that they rely on statistical information of the noise. As detailed in Section II, in order to accurately capture the statistical correlation between different nodes, the number of samples increases with the number of nodes. With hundreds of thousands or even millions of nodes present in the power grids, a huge number of samples are needed and the statistical modeling can hardly be accurate. It is imperative to search for alternative methods that can work even with limited number of samples.

Machine learning algorithms have achieved extraordinary improvement in various applications, which include image searching [9], action recognition [10], and even beating a human champion at Go [11]. Nowadays they are also applied to the field of electronic design automation, including high level synthesis, physical design, circuit test, online thermal/power management and so on [12]–[15]. In this paper, we attempt to make use of the newest breakthrough, generative adversarial network (GAN), to solve the problem of noise sensor placement. Specifically, the proposed method uses a limited number of noise samples obtained from power grid simulation to train GAN and extract critical features, from which a large number of samples can quickly be yielded for optimal sensor placement through an efficient heuristic method. Experimental results on an industrial design show that compared with a simple heuristic method which takes in the same number of samples, our approach can reduce the miss rate of voltage emergency detection by up to 65.3%.

The remainder of this paper is organized as follows. We first present the background about the on-chip noise sensor placement problem with its formulation and related work in Section II. The GAN scheme adopted is introduced in Section III with our proposed strategy to decide locations for noise sensors. Experimental results are presented in Section IV and concluding remarks are given in Section V.

[1] A noise map is formed by the noise values of all the nodes in the power grid at a particular time, which can be obtained by transient simulation.

978-1-5386-4034-0/17/$31.00 ©2017 IEEE

II. Background and Motivation

Following [4], the on-chip noise sensor placement problem can be formally defined as follows. Given the following information as inputs

1) M candidate nodes in the power grid for noise sensor placement and their respective simulated noise information
2) voltage threshold t for the voltage emergencies, which is specified by the designer and is the same for all the sensors;
3) total number of noise sensors n to be placed.

Our objective is to find the optimal n locations out of the M candidate nodes so that the miss rate, defined as the probability that a voltage emergency occurred somewhere on chip but none of the placed sensors capture it, is minimized. The miss rate can be mathematically defined as

$$\begin{aligned} Miss\ Rate =& P(Z_{max} \geq t | Z_{r_1} \leq t, Z_{r_2} \leq t, \cdots, Z_{r_n} \leq t) \\ =& P(Z_{max} \geq t | max(Z_{r_1}, Z_{r_2}, \cdots, Z_{r_n}) \leq t) \end{aligned}$$

where Z_{max} is the maximum noise among all the nodes in the power grid, $Z_{r_1}, Z_{r_2}, \cdots, Z_{r_n}$ are noise values at the n nodes where the sensors are connected, and t is the voltage threshold. If there is a voltage emergency somewhere on chip, at least one node must have $Z_i \geq t, 1 \leq i \leq M$, which infers $Z_{max} \geq t$.

Originally, a method named Eagle-Eye is proposed based on statistical noise analysis in [4]. The noise of on-chip node i, either Gaussian or non-Gaussian, can be represented as follows

$$Z_i = F_i(\boldsymbol{X}) = H_i(\boldsymbol{G}) + \Delta R_i \tag{1}$$

where $\boldsymbol{X}$ is a set of common correlated factors that result in the variation of voltage noise through function F_i. Through modelling techniques, the noise can be represented by function $H_i(\boldsymbol{G})$, where $\boldsymbol{G}$ is an m-dimensional uncorrelated random variable that models the global variation sources (common for all positions) which can be extracted from $\boldsymbol{X}$ through either principle component analysis (PCA) for Gaussian or independent component analysis (ICA) for non-Gaussian distributions of $\boldsymbol{X}$. This transformation is based on the covariance matrix estimation. In addition, ΔR_i models the independent source of noise variation specific to position i. After this, Sensing Quality Metric (SQM) for a set of positions is defined as the probability that the maximum noise among them exceeds the voltage threshold. Based on SQM, noise sensor placement is decided by existing algorithms, since it can be interpreted as the max set cover (MSC) problem [16].

Eagle-Eye is built on the basis of transformation in (1), which is calculated from covariance matrix estimation. Comrey and Lee [17] urged researchers to obtain 500 or more samples per variable. Therefore, even for a medium scale power grid with 100,000 nodes, the number of noise samples required is 50 million. SPICE simulation to get such a huge number of samples will take an extremely long time. In this paper, we will try to address this scalability problem by devising a generative adversarial network based approach.

III. GAN Based Noise Sensor Deployment

The proposed method to decide where to place noise sensors consists of two major parts. One is to efficiently produce more noise maps via GAN, and the other is to determine optimal deployment for noise sensors from the produced noise maps.

A. Produce noise maps based on GAN

GAN was developed by Goodfellow et al. as a framework to train a generative model by an adversarial process [18]. Recent applications of GANs have shown that they can produce excellent image samples [19], [20]. A well-trained GAN enables fast sampling from the learned distribution of images. That is, if noise maps are treated as images, based on limited training noise maps, GAN can produce a number of new noise maps with the same distribution as the training ones. This makes it an ideal method to generate a large number of noise maps.

The basic idea of GAN is to set up a game between two players. One player named discriminator examines samples and tries to distinguish the real samples from training dataset from the synthetic 'fake' samples, which are noise maps here. The other player named generator produces such 'fake' noise maps to 'fool' the discriminator. The competition in the game drives both players to improve their skills and finally become experts on their own tasks.

Formally, the two players in the game are represented by two functions: a discriminator network D and a generator network G. Both functions are realized by deep neural networks and are differentiable with respect to their inputs and parameters. As shown in Fig. 1, the generator network G maps random variable $z \sim p(z)$ to data space $x = G(z)$. The discriminator network D assigns a probability $p = D(x) \in [0, 1]$ which represents the probability that x comes from the real data rather than the generator's distribution p_g.

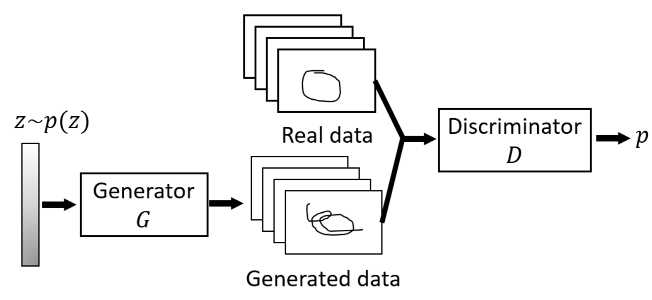

Fig. 1. Generative adversarial networks

The solution to this game is a Nash equilibrium where each function reaches a local minimum of its loss function with respect to its parameters. It has been proved that, given both generator and discriminator have enough capability, the only global optimum is $p_g = p_{data}$, where p_g is the implicit probability distribution of the samples $G(z)$ obtained when $z \sim p(z)$ and p_{data} is the real probability distribution of data.

Since the input for GAN is usually a set of images with pixels ranging from 0 to 255, noise maps need slight transformation before being fed to GAN. As mentioned in Section II, there are M candidate nodes for a chip, thus a noise map can be formulated as a $1 \times M$ vector of noise values. We can take the noise values of every position as a pixel. In simulated noise maps, each pixel varies in a small range from 0 to 0.3 with Volt as unit in our record. Therefore, to make the full use of space provided by image domain, every element of noise maps is scaled to the range from 0 to 255 as pixels. After scaling, noise maps can be treated as images to be fed to GAN directly. Accordingly, the threshold that should be recognized as a voltage emergency can be defined by a proportion of the maximum voltage, whose value is 255 after scaling, instead of a concrete value based on experience in other works [4]. As a result, the voltage threshold $t = p \times Max$, where $Max = 255$ and p is the proportion selected.

Since the most critical problem of the original GAN concept is the non-convergence, the deep convolution generative adversarial network (DCGAN), an extension of GANs, makes it more stable to train in most settings through utilizing several tricks. Therefore, DCGAN instead of original GAN is adopted in implementation of our experiments. After training process, the generator network G of DCGAN should be able to produce noise maps with the same distribution as the simulated noise maps plus variation.

B. Efficient selection

To deploy on-chip noise sensors properly, we propose an efficient algorithm based on the produced noise maps through DCGAN, which is named as efficient selection (ES). It can skip statistical noise analysis that is too complex to implement. The key idea is to remove samples as long as they can be detected by any existing noise sensor, since they cannot provide further information for subsequent noise sensor placement. Details of efficient selection are shown in Algorithm 1.

For inputs, noise map matrix M_q is a $N \times M$ matrix of quantized produced noise maps. N is the number of noise maps and M is the number of candidate nodes. What is different from definitions in Section III-A is that every element in M_q is quantized by threshold t so that if a voltage noise value is beyond the threshold, it is quantized as 1; otherwise it is 0. In addition, the number of noise sensors n to be placed is the same as defined in Section II. As for output, S_{noise} is a list containing selected positions to place noise sensors.

In Algorithm 1, sum(M_q) returns the result of summing every element in M_q, while sum(M_q, axis=0) returns a vector whose each element is the sum of every column of M_q. Later on, max(sum_c) returns the maximum value in sum_c. $M_q.rows$ returns the number of rows remaining in matrix M_q. When sum(M_q)==0 or $M_q.rows$==0, there is no noise remaining in M_q or there is no more available noise map to suggest any position for noise sensors. In addition, idx_c is the position where voltage emergency is most likely to happen and it is added into S_{noise} through the operation S_{noise}.append(idx_c). After that, any noise map, that is, any

Algorithm 1 Efficient selection

Input: Quantized noise map matrix $\boldsymbol{M_q}$.
    Number of noise sensors $\boldsymbol{n}$.
Output: Positions to place noise sensors S_{noise}.
for $i = 1$ to n **do**
  if sum(M_q)==0 or $M_q.rows$==0 **then**
    All noise covered!
    return S_{noise}
  end if
  sum_c = sum(M_q, axis=0)
  max_c = max(sum_c)
  idx_c = column number where max_c locates
  S_{noise}.append(idx_c)

  //remove noise maps that can be detected
  //by the sensor placed on idx_c
  remove any row r in M_q that r(idx_c) = 1
end for
return S_{noise}

row r in M_q, is not able to provide more useful information for deployment, if it gets 1 at the idx_c position. Thus, such rows corresponding to those samples are removed afterwards.

IV. Experimental results

The experiments are performed on an industrial design for transient analysis. The simulation is conducted on PowerRush [21]–[23], while the DCGAN implementation is based on [24]. Simulated noise maps are sequentially and randomly divided into training set and test set. The training set is fed to DCGAN to produce more noise maps, while the test set is used to measure the miss rate of voltage emergency detection after sensor placement.

The simulation to generate enough samples for Eagle-Eye requires a long time and huge memory to complete, as Eagle-Eye is not very suitable for large scale power grids. Accordingly, in this paper we directly compare the proposed approach with a new yet intuitive heuristic method. In the heuristic method, the n-top positions with highest voltage emergency occurrence frequency in the test set are selected. The number n here is the same as the number of sensors to be placed as mentioned in Algorithm 1.

Fig. 2 displays comparison between the efficient selection based on GAN and the heuristic method based on training set in terms of miss rate of voltage emergency detection versus different number of noise sensors employed. In this figure, x-axis corresponds to the number of noise sensors placed, and y-axis is the corresponding miss rate of voltage emergency detection. Since a chip cannot afford many noise sensors because of overhead [4], in our experiments the number of noise sensors is nine at most.

From Fig. 2, we can see that the efficient selection based on GAN (the line named by "ES based on GAN" in Fig. 2) outperforms the heuristic method based on the training set, which decreases the miss rate by 65.3% when the number of noise sensors is eight. This is because the trained GAN provides additional samples following the same noise distribution, allowing more accurate placement of the noise sensors. There are some cases, however, where ES based GAN does not show significant improvement over the heuristic, and this is probably because the samples in the training set already contain enough statistical information to guide the placement of noise sensors, and adding additional samples from GAN does not provide any further help.

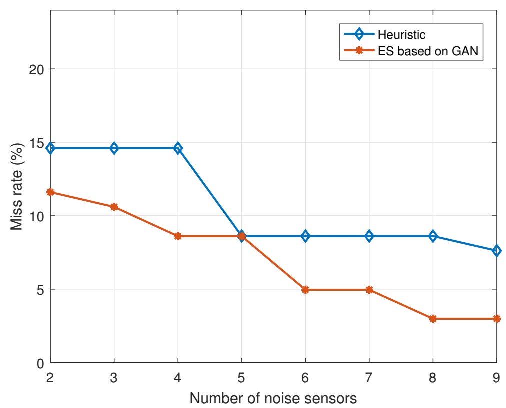

Fig. 2. Miss rate vs. the number of on-chip noise sensors employed in different methods

V. Conclusions and Future Work

The near-threshold computing paradigms have been used widely, because of the relentless efforts towards power reduction of integrated circuits. It leads to the significantly reduced noise margin, and it is no longer possible to fully assure power integrity at design time. Therefore, designers have proposed various runtime techniques to manage voltage emergencies based on noise sensors on chips. There exist some related works for placement of noise sensors, but they all exploited the statistical modeling of noise. However, for large scale power grids with a high dimensional space that requires a large number of samples, it will take very long simulation time to get enough samples. Thus, these works may not work anymore. In this paper, a novel approach based on GAN is proposed, which only requires a small number of samples. Experimental results show that compared with the heuristic method which takes in the same number of samples, our approach can reduce the miss rate by up to 65.3% on an industrial design.

Acknowledgment

This work was supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre “Invasive Computing” (SFB/TR 89).

References

[1] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, “Near-threshold computing: Reclaiming moore’s law through energy efficient integrated circuits,” *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[2] A. Muhtaroglu, G. Taylor, and T. Rahal-Arabi, “On-die droop detector for analog sensing of power supply noise,” *IEEE Journal of solid-state circuits*, vol. 39, no. 4, pp. 651–660, 2004.

[3] V. J. Reddi, M. S. Gupta, G. Holloway, G.-Y. Wei, M. D. Smith, and D. Brooks, “Voltage emergency prediction: Using signatures to reduce operating margins,” in *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*. IEEE, 2009, pp. 18–29.

[4] T. Wang, C. Zhang, J. Xiong, and Y. Shi, “Eagle-eye: a near-optimal statistical framework for noise sensor placement,” in *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 437–443.

[5] T. Wang, C. Zhang, J. Xiong, and Y. Shi, “On the deployment of on-chip noise sensors,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 4, pp. 519–531, 2014.

[6] T. Wang, C. Zhang, J. Xiong, P.-W. Luo, L.-C. Cheng, and Y. Shi, “On the optimal threshold voltage computation of on-chip noise sensors,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1744–1754, 2016.

[7] T. Wang, C. Zhang, J. Xiong, P.-W. Luo, L.-C. Cheng, and Y. Shi, “Variation aware optimal threshold voltage computation for on-chip noise sensors,” in *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 205–212.

[8] X. Liu, S. Sun, X. Li, H. Qian, and P. Zhou, “Machine learning for noise sensor placement and full-chip voltage emergency detection,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 3, pp. 421–434, 2017.

[9] C. Rosenberg, “Improving photo search: A step across the semantic gap,” 2013.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[12] J. Liu, D.-C. Juan, and Y. Shi, “Effective cad research in the sea of papers,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2015, pp. 781–785.

[13] B. Yu, D. Z. Pan, T. Matsunawa, and X. Zeng, “Machine learning and pattern matching in physical design,” in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*. IEEE, 2015, pp. 286–293.

[14] L.-C. Wang, “Data mining in functional test content optimization,” in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*. IEEE, 2015, pp. 308–315.

[15] H. Hantao, P. S. Manoj, D. Xu, H. Yu, and Z. Hao, “Reinforcement learning based self-adaptive voltage-swing adjustment of 2.5 di/os for many-core microprocessor and memory communication,” in *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 224–229.

[16] U. Feige, “A threshold of ln n for approximating set cover,” *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[17] S. P. Reise, N. G. Waller, and A. L. Comrey, “Factor analysis and scale revision.” *Psychological assessment*, vol. 12, no. 3, p. 287, 2000.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[20] E. L. Denton, S. Chintala, R. Fergus *et al.*, “Deep generative image models using a laplacian pyramid of adversarial networks,” in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[21] J. Yang, Z. Li, Y. Cai, and Q. Zhou, “Powerrush: A linear simulator for power grid,” in *Proceedings of the International Conference on Computer-Aided Design*. IEEE Press, 2011, pp. 482–487.

[22] J. Yang, Z. Li, Y. Cai, and Q. Zhou, “Powerrush: Efficient transient simulation for power grid analysis,” in *Proceedings of the International Conference on Computer-Aided Design*. ACM, 2012, pp. 653–659.

[23] J. Yang, Z. Li, Y. Cai, and Q. Zhou, “Powerrush: An efficient simulator for static power grid analysis,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2103–2116, 2014.

[24] T. Kim. (2016) DCGAN-tensorflow. [Online]. Available: https://github.com/carpedm20/DCGAN-tensorflow