15-4

Real-Time TCAD: a new paradigm for TCAD in the artificial intelligence era

Sanghoon Myung*, Jinwoo Kim, Yongwoo Jeon, Wonik Jang, In Huh, Jaemin Kim, Songyi Han,
Kang-hyun Baek, Jisu Ryu, Yoon-Suk Kim, Jiseong Doh, Jae-ho Kim, Changwook Jeong, Dae Sin Kim

Computational Science and Engineering Team, Data and Information Technology Center,
Device Solution Business, Samsung Electronics Co., Ltd., Gyeonggi-do 18448, Korea.
* Email Address: shoon.myung@samsung.com

***Abstract-* This paper presents a novel approach to enable real-time device simulation and optimization. State-of-the-art algorithms which can describe semiconductor domain are adopted to train deep learning models whose input and output are process condition and doping profile / electrical characteristic, respectively. Our framework enables to update automatically deep learning models by estimating the uncertainty of the model prediction. Our Real-Time TCAD framework is validated on 130nm processes for display driver integration circuit (DDI), and 1) prediction time was 530,000 times faster than conventional TCAD, and time spent for process optimization was reduced by 300,000 times compared to human expert, 2) the model achieved average accuracy of 99% compared to TCAD simulation results, and thus, 3) process development time for DDI was reduced by 8 weeks.**

Keywords—TCAD, neural network (NN), convolutional neural network (CNN), recurrent neural network (RNN), Real-time prediction, Real-time optimization, active learning.

I. Introduction

Over the past few decades, the number of transistors in an integrated circuit (IC) has doubled every two years, as predicted by the Moore`s law. As the size of transistors scales down, predicting non-ideal phenomena, e.g., short-channel effects, becomes important. Generally, engineers rely on the technology computer-aided design (TCAD) simulators to predict and solve these complex physical phenomena [1]. Although TCAD simulators require proper calibration of the model parameters to match simulation results with measurements, they have been successful so far especially in terms of predicting the performance of transistors. However, due to several hours of simulation time, the role of TCAD has been limited to predict transistor performance in the research and development phase, not in the mass production phase. Recently, to address this problem, adopting artificial intelligence (AI) technologies to decrease TCAD simulation turn-around-time (TAT) has been investigated. One of the advantages of employing AI in the manufacturing industry is that it opens a possibility of fast decision making without human intervention, which in turn, can significantly shorten the development cycle. However, current AI studies for manufacturing are not mature enough to realize fully-automated decision making due to the following reasons: First, low model prediction power, most of previous studies[2-4] use some shallow learning (SL) such as artificial neural network, random forest, Gaussian process and so on. One of the limitation of SL models is that they are difficult to interpret the relationship of output values. For instance, SL cannot accurately predict the current-versus-voltage characteristics curves, whose current levels (outputs) are closely correlated with each other, according to semiconductor device theory. Concrete example will be presented in Section III. Second, trustworthiness problem of the model, it is important to avoid potential risks of blindly trusting the result of the models by checking predictive uncertainty. However, previous studies cannot catch this problem because they focused on improving the performance of the model.

In this paper, we present a novel framework, called Real-Time TCAD (RTT), to enable the automated training and decision making. The contribution of this paper is mainly threefold: First, we proposed the novel algorithm combining recurrent neural network (RNN) / convolutional neural network (CNN) architecture in semiconductor domain. Second, the novel approach to enable automatic update of DL models and to avoid the risks by using calculated uncertainty. Last, we experimentally showed that our RTT model achieves the average accuracy 99% compared to TCAD simulation, and prediction and optimization time is ~ 530,000 times faster and 300,000 times faster than conventional TCAD and human expert, respectively. Process development time for DDI was reduced by 8 weeks.

II. Background

A. TCAD Setup

Before illustrating our RTT framework, TCAD simulation 130nm process will be briefly introduced. This process is well-calibrated so that TCAD results show great agreement with measurement data. Simulation dimension is shown in Fig. 1 and the square indicated by dashed line is RTT-process model's output dimension. The input variable consists of 10 variables associated with structure and 24 variables of ion implantation. The process simulator receives input variables as zero-dimensional (0D) scalar values and solve dopant implantation, diffusion, and activation phenomena to calculate final doping profile of MOSFET as a two-dimensional (2D) image. The device simulator receives same input values as the process simulator, and solves electron transport to calculate *I*-*V* curves as one-dimensional (1D) arrays, and 0D scalar values such as threshold voltage (V_T), breakdown voltage (BV) and etc.

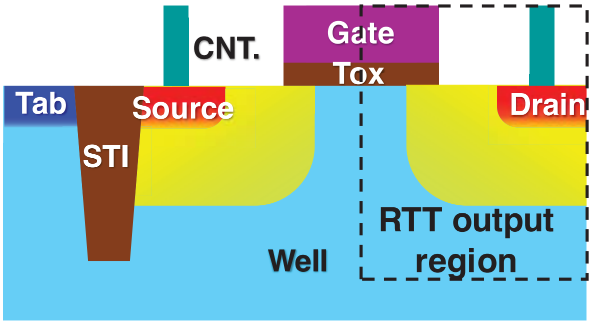


Fig. 1. Simulation structure. Dashed line area is a simulation

B. Pefromance Metric for RTT Process Model

To assess the performance of RTT-process model, Structure SIMilarity (SSIM) and intersection over union (IOU) are used. SSIM is a widely used metric for image quality assessment in the computer vision field [5]. By examining doping profile images with different SSIM values, minimum SSIM required for the RTT-process model is set to 0.94, although for images of large luminance and contrast, >0.96 is required to ensure good image quality [6]. Since SSIM alone cannot capture important regions of MOSFET doping profiles, such as n/p-type junction areas, IOU of important regions (Fig. 2 (a)) is also used. The minimum IOU required is set to 0.9 since the area should match at least 95% (Fig. 2 (b)). Finally, average of SSIM and IOU are used as a score of RTT-process models and the target score is ≥0.92.

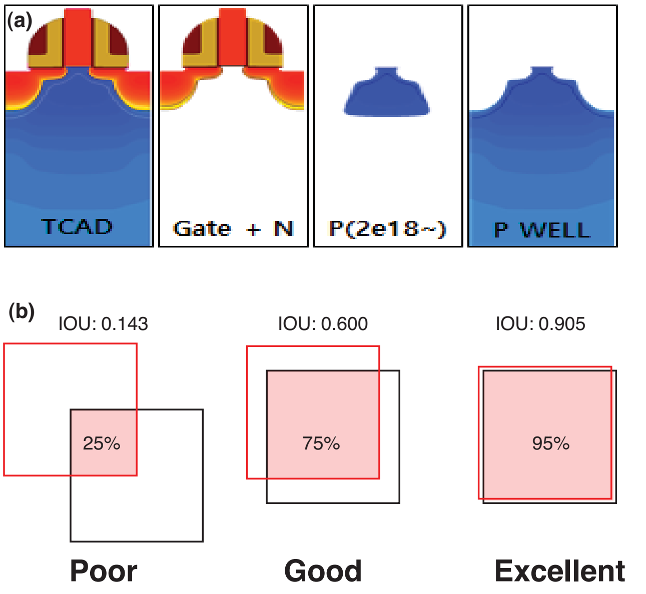


Fig. 2. (a) Classification of regions by doping type. (b) An example of IOU for square boxes.

C. Pefromance Metric for RTT Device Model

For RTT-device model assessment, we introduce mean-absolute-percentage-accuracy (MAPA) as follows:

$$\mathrm{MAPA} = \frac{1}{N}\sum_{i=1}^{N}\left(1-\left|\frac{Y_i-\bar{Y}_i}{Y_i}\right|\right) \qquad (1)$$

where Y and $\bar{Y}$ are true and prediction value, respectively and N is the number of samples. However, MAPA alone cannot capture sensitivity discrepancy between true and prediction values. Hence, we additionally introduce the coefficient of determination (R-squared), which measures how much the model predicted values statistically matches with the true values. To take advantage of both metrics, average of MAPA and R-squared is used as a model performance score of RTT-device models and the target score is set to ≥0.98.

III. Methodology

A. Preliminary

Generally, TCAD-based semiconductor process optimization is done in a heuristic and iterative manner. For the optimization, firstly TCAD simulations for the process are set up and then iterative adjustment of process conditions until target device characteristics are met. Since a single simulation takes several hours, usually the performance optimization takes several days to a few weeks. This TAT limits the use of TCAD-based optimization results in a mass production phase. To be useful in mass production phase, simulation TAT and error should be within a few seconds and less than 1%, respectively. By adopting deep ensemble approach [7], continual addition of TCAD data and re-training of the models can be fully automated in our framework (Fig. 3).

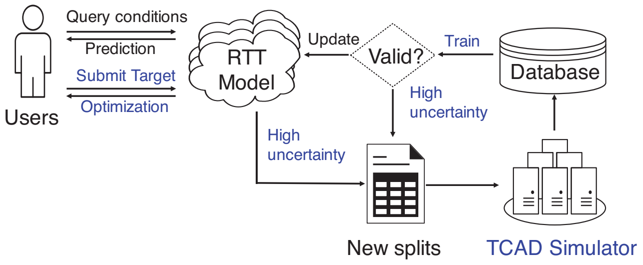


Fig. 3. Illustraion of RTT framework.

B. RTT for Process Simulation

TCAD process simulator takes process conditions as inputs, solves continuum physical models, and outputs distribution of doping concentration as images. Since the output of RTT process model is an image, it is natural to adopt CNN structure for the model [8]. From a baseline CNN structure, several algorithm engineering techniques are applied to enhance the model performance. First of all, the residual block, introduced in ResNet [9], is added to prevent vanishing gradients and ensure efficient training of the model. Secondly, swish activation is used in order to enhance training accuracy [10]. Last but not least, Group Normalization (GN) [11] is also applied. Normalization techniques make training loss landscape smoother and thus enable fast and stable training of deep learning models [12]. Although batch normalization [13] is the one of the most famous for normalization technique, GN is used for the model to achieve stable training for small batch size. In semiconductor process data, it is important to capture critical regions and dimensions, such as oxide thickness and effective channel length, but conventional CNN structures are not good at processing positional information of images [14]. Thus, to boost the model's ability to capture important regions by embedding positional information of data, we added coordinate convolution which adds Cartesian coordinate for every convolution. As shown Fig. 4, the final score of the model is 0.96, which is above our target score, 0.92.

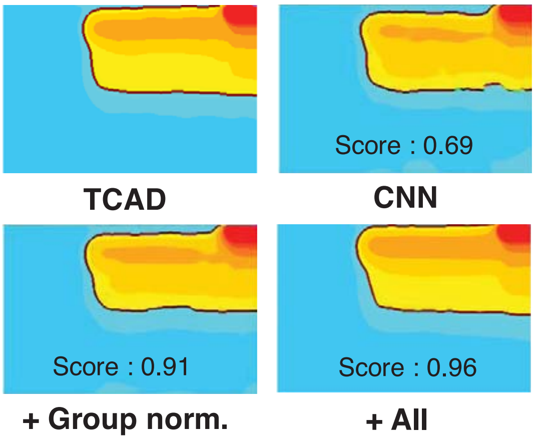


Fig. 4. An example RTT process model output image as model engineering techniques are applied. The score is an average of SSIM and IOU.

C. RTT for Device Simulation

Device simulator takes process condition as inputs, and calculates electron transport and then, generates electrical characteristics such as I-V curves, BV, VT as outputs. Most preliminary work to predict such characteristics with AI has been developed using SL, which cannot catch relationship between outputs. For example, let's say one of test data is out of training range (Fig. 5 (a)). For SL, the model needs to extrapolate to predict out-of-training-range test data, while when relationships of the point with its neighbors are considered, the point may not be regarded as out-of-training range. To capture relationships between neighboring points, we adopted a RNN structure, which is specially designed to learn sequential data, to RTT device models. In a RNN structure, the current is predicted from zero to max gate (drain) voltage sequentially for a fixed drain (gate) voltage (Fig. 5 (b)). As applying the RNN structures with Layer Normalization (LN) [15], the model score increases from 0.62 to 0.89. In addition, from domain knowledge on device physics, current values of neighboring drain and gate voltages are highly correlated. Hence, we adopted a 2D CNN structure to capture relations between neighboring gate bias and drain bias at the same time. For the precise prediction, an accurate calculation of derivatives of current with respect to voltage, is essential. Thus, in order to ensure RTT device model learns both current and trans-conductance values simultaneously, we used average of mean-squared-error and derivative losses. The device model cannot be predict proper G_M without the derivative loss even when it can accurately predict *I*-*V* curves (Fig. 6 (a)). The final score of device model is 0.99 (Fig. 6 (b)).

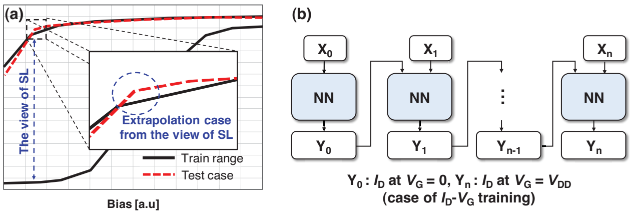


Fig. 5. (a) Example test data to illustrate the weakness of SL. (b) RNN structure for RTT device model.

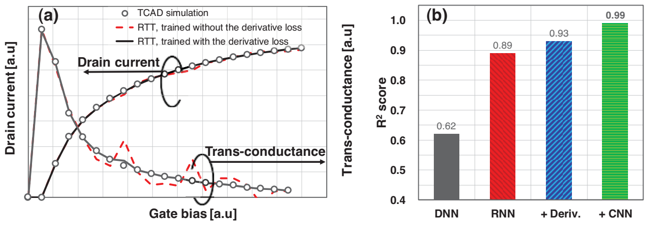


Fig. 6. (a) I_D and G_M-*versus*-V_G curve whether applying derivative loss or not. (b) RTT device model score as model structure engineering is applied.

D. Optimization

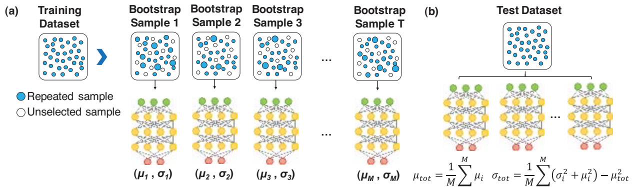
Once RTT models are accurately trained, they can be directly used to find the optimal process condition to achieve target device performances. For this task, optimization algorithms such as genetic algorithm (GA), simulation annealing (SA), gradient descent (GD) are commonly used. In general, GA and SA are better at finding global optimum but requires more iterations than GD. Since RTT model is based on neural network, which is easily differentiable, 10,000 GD from different Latin hypercube sample points [16] are executed to find global optimum in real-time. By comparing the optimization results with the results obtained by GA, it is confirmed that the optimal point found is global optimum. Regarding TAT, our model found the global optimum within 60 seconds, which is 280 times faster than GA.

E. Confidence Interval Prediction

One of the well-known problem of deep learning models is that it is hard to tell whether the model prediction is trustable or not. When the model is asked to predict electrical characteristics of unseen process conditions, due to not enough train data or queried data out-of-training range, it is desired that the model outputs prediction results with low confidence rather than completely wrong prediction results with high confidence. To do that, we changed loss function which learns confidence interval, assumed by multivariate Gaussian process [17]. The loss function is negative-log-likelihood (NLL):

$$NLL = \frac{(y-\mu(x))^2}{2\sigma^2(x)} + \frac{\log\sigma^2(x)}{2}, \qquad (2)$$

where $\mu(x)$ and $\sigma^2(x)$ are mean and variance of model outputs. The numerator of the first term of eqn. (2) is introduced to reduce the gap between the mean of predicted values and the truth, i.e. it is same as mean-squared-error. If the mean-squared-error is relatively large, the model is likely to have large variance values to decrease the first term of the loss. But since the increase of the variance increases the second term of the loss, the model learns an appropriate variance values balancing between two terms. However, a single model tends to over-fit to the train samples, so to overcome overfitting issue, ensemble method is applied [7]. In training phase, each model is trained with different samples generated by bootstrap algorithm (Fig. 7 (a)). In the testing phase, the predicted outputs of the models are the mean and variance of the ensemble mixture (Fig. 7 (b)). When queried data point is in the training range, confidence interval of *I*-*V* curve is negligible so engineers can trust the prediction and performs experiments based on the prediction results (Fig. 8 (a)). For queried data point out-of-training range, since the model can tell the queried point is unseen during the training, predicted confidence interval is significantly large, as expected (Fig. 8 (b)). After RTT model is automatically retrained with previously unseen data points, which is obtained by TCAD simulations, the RTT model accurately predicts *I*-*V* curve with low confidence interval (Fig. 8 (c)).



$$\mu_{tot} = \frac{1}{M}\sum_{i=1}^{M}\mu_i \quad \sigma_{tot} = \frac{1}{M}\sum_{i=1}^{M}(\sigma_i^2 + \mu_i^2) - \mu_{tot}^2$$

Fig. 7. (a) The models are independently trained using bootstrap techniques in training phase. (b) The ensemble prediction as Gaussian whose mean and variance are the mean and variance of the mixture, respectively in test phase.

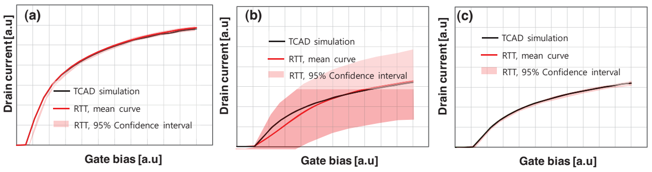


Fig. 8. Prediction uncertainty plot (a) in training range (b) out-of-training range (c) after retraining out-of-training range sample.

IV. Experiment Results

A. RTT Models's Performance

TCAD simulator which is calibrated with 130 nm DDI process are used to generate TCAD data to train and evaluate the proposed RTT model. We generated 1,050 TCAD simulation samples with changing input variables. Among them, 1,000 samples are used in the model training phase, and the remaining 50 samples are used to evaluate the performance of trained model. The evaluated test score of the process model and the device model are equal to 0.96 and 0.99, respectively. Prediction times of the RTT process and device models are 60 and 0.1 seconds, respectively, while them of TCAD process and device simulations are ~1,200 and ~530,000 times longer, respectively.

B. Process Optimization Results

Next, the RTT device model is used to optimize actual semiconductor process. Our targets are listed in Table I: while keeping V_T as close to the target value as possible, I_{ON}, BV_{OFF}, and BV_{ON} are (I_{OFF} is) required to be maximized (minimized). With the RTT model, optimal process condition can be searched within 50 seconds, which is 300,000 times faster than the human expert (Fig. 9 (a)). Also, it is confirmed that at the searched optimal condition, the results of RTT device model agree with them of the TCAD simulation results with 99% accuracy (Table Ⅵ and (Fig. 9 (b))). Finally, our optimal condition has resulted in 11.4% gain of performance on average compared to the reference condition of the process. With the assist of RTT model optimization, process development time for DDI is reduced by 8 weeks.

TABLE I. Optiziation Results.

	V_T	I_{ON}	I_{OFF}[b]	BV_{OFF}	BV_{ON}
Target[a]	1	> 1	< 1	> 1	> 1
Exp.[c]	0.98	0.99	0.33	1.06	N/A
Opt.	1.04	1.07	0.56	1.09	1
TCAD	1.02	1.08	0.6	1.09	1
Acc.	98.4%	99.7%	98.1%	100%	99.6%

[a] Numbers are normalized by target values for confidential reasons.

[b] For I_{OFF}, $\log(I_{OFF})$ is used for accuracy calculation.

[c] Exp. are experimental results on 130nm DDI process.

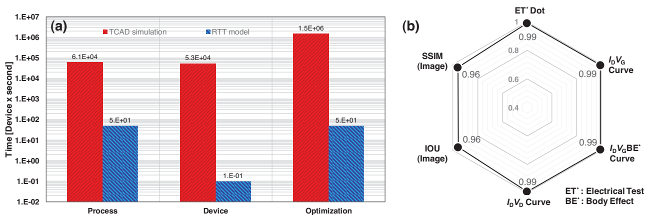


Fig. 9. (a) Prediction time for each task. (b) Performance score of RTT models on TCAD simulation results at optimal process condition.

V. Concolusion

We propose a new approach to realize real-time TCAD prediction and optimization for semiconductor device designs. For 130nm DDI, simulation time is 530,000 times faster than conventional TCAD simulators, and the process development time is reduced by 8 week.

The whole procedure described here, from TCAD simulation data generation to trained model deployment and process optimization, can be automated to realize fully automated technology design.

References

[1] Dennard, Robert H., et al. "Design of ion-implanted MOSFET's with very small physical dimensions." IEEE Journal of Solid-State Circuits 9.5 (1974): 256-268.

[2] Shi, Mengchao, Pinghui Mo, and Jie Liu. "Deep neural network for accurate and efficient atomistic modeling of phase change memory." IEEE Electron Device Letters 41.3 (2020): 365-368.

[3] Chen, Jing, et al. "Powernet: SOI Lateral Power Device Breakdown Prediction With Deep Neural Networks." IEEE Access 8 (2020): 25372-25382.

[4] Carrillo-Nuñez, Hamilton, et al. "Machine learning approach for predicting the effect of statistical variability in Si junctionless nanowire transistors." IEEE Electron Device Letters 40.9 (2019): 1366-1369.

[5] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612.

[6] Bartolini, Andrea, Martino Ruggiero, and Luca Benini. "Visual quality analysis for dynamic backlight scaling in LCD systems." 2009 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2009.

[7] Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." Advances in neural information processing systems. 2017.

[8] Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[9] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[10] Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. "Searching for activation functions." arXiv preprint arXiv:1710.05941 (2017).

[11] Wu, Yuxin, and Kaiming He. "Group normalization." Proceedings of the European conference on computer vision (ECCV). 2018.

[12] Santurkar, Shibani, et al. "How does batch normalization help optimization?." Advances in Neural Information Processing Systems. 2018.

[13] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

[14] Liu, Rosanne, et al. "An intriguing failing of convolutional neural networks and the coordconv solution." Advances in Neural Information Processing Systems. 2018.

[15] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450 (2016).

[16] Tang, Boxin. "Orthogonal array-based Latin hypercubes." Journal of the American statistical association 88.424 (1993): 1392-1397.

[17] Nix, David A., and Andreas S. Weigend. "Estimating the mean and variance of the target probability distribution." Proceedings of 1994 ieee international conference on neural networks (ICNN'94). Vol. 1. IEEE, 1994.