Inverse Design of FinFET SRAM Cells

Rui Zhang[a], Zhaocheng Liu[a], Kexin Yang[a], Taizhi Liu[a], Wenshan Cai[a], and Linda Milor[a, *]
[a] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*
* linda.milor@ece.gatech.edu

***Abstract*—A convenient method based on deep neural networks and an evolutionary algorithm is proposed for the inverse design of FinFET SRAM cells. Inverse design helps designers who have less device physics knowledge obtain cell configurations that provide the desired performance metrics under selected wearout conditions, such as a set specific stress time and use scenario that creates a specific activity level (duty cycle and transition rate). The cell configurations being considered consists of various process parameters, such as gate length and fin height, in the presence of variations due to process and wearout. The front-end mechanisms related to wearout include negative bias temperature instability (NBTI), hot carrier injection (HCI), and random telegraph noise (RTN). The process of inverse design is achieved quickly and at good accuracy.**

Keywords—Inverse Design, FinFET, SRAM, Performance Metrics, Wearout Mechanisms, NBTI, HCI, RTN

I. Introduction

Small area, low power consumption, and superior performance are always pursued for the design of an advanced SRAM. With the increase of requirements on system capacity, technology scaling of conventional CMOS is invented to achieve more complicated circuits under chip area constraints. Scaling leads to more leakage power and performance degradation. The design targets are difficult to achieve due to the drastic increase of sub-threshold leakage which leads to unacceptable leakage currents and power consumption. The leakage is due to short channel effects (SCEs). In general, we can maintain device/circuit performance in an acceptable range through mature processes with thinner gate oxides and high-k dielectric materials. In the deeply scaled regime, fin field-effect transistors (FinFETs), such as double-gate FETs and trigate FETs, are proposed to overcome the disadvantages, such as SCEs, with their better gate control capability [1]-[4]. FinFET characteristics are related to various process conditions and design parameters, such as gate length, fin width, fin height, doping concentration, fin pitch, *etc*. Variation of process conditions affects the device characteristics, such as the I_{on}/I_{off} ratio, threshold voltage, and leakage current might change substantially. Since there are so many manageable process parameters at the device level, finding a suitable device configuration for circuit design is labor-intensive work if the designers are not sure about the importance of every process parameter. We propose a solution which combines neural networks and an evolutionary algorithm to overcome this problem.

Meanwhile, deeply scaled FinFETs are sensitive to time-zero variability and front-end/back-end wearout mechanisms, such as negative bias temperature instability, hot carrier injection, random telegraph noise, gate oxide breakdown, electromigration, and stress induced voiding [5]-[15]. The front-end wearout mechanisms affect device parameters and cause a shift in FinFET SRAM performance metrics. The back-end wearout mechanisms affect interconnect quality and lead to its breakdown. To obtain a high-end SoC, with the best possible area, power, and performance, it is important to find an optimized solution for the FinFET SRAMs based on specific goals on the performance metrics, such as static noise margin (SNM), minimum VDD for data retention (VDD-ret-min), read delay & power, write delay & power, and leakage power, while accounting for process parameters and wearout mechanisms. In this study, degradation and variation due to the front-end wearout mechanisms, namely NBTI, HCI, and RTN, are taken into consideration.

SRAM cells' performance metrics can be checked with trials in Spice simulations. However, since the number of adjustable device parameters is large and the initial design needs physical insights and intuitive reasoning, the design process needs a rich knowledge of SRAMs to exploit the full parameter space. Benefiting from the rapid development of artificial intelligence (AI), some science problems which require human perception can be solved with AI algorithms. The methods based on deep neural networks (DNNs) have been incorporated into the discovery of nanophotonics and to speed-up the design process in [16]. In [17] and [18], machine learning techniques are applied to speed up the modeling and simulation of circuits for fast simulation and IP protection with good accuracy. These studies verified the good efficiency and comprehensiveness of the application of machine learning techniques. In this study, we propose a solution to circumvent the conventional design procedure for FinFET SRAM cells by using deep neural networks and an evolutionary algorithm. When fed with inputs of the customer-defined performance metrics, wearout status and use scenario, the constructed algorithm generates a cell configuration which provides the requested performance metric with high confidence. The process is the inverse design of FinFET SRAM cells which can alleviate a designer from the complicated explorations in a huge domain of device parameters. Inverse design can be accomplished in several seconds. Meanwhile, the explored device parameters can ensure the worst case of the performance metrics meet design requirements when variations are considered.

This paper is arranged as follows. Section II gives detailed descriptions for inverse design, the evolutionary algorithm, and deep neural networks. Meanwhile, the models for front-end wearout mechanisms are introduced. In Section III, the application of the proposed solution is shown with several examples. The inverse design for FinFET SRAM cells in the fresh state and the degraded states with specific stress times and specific combinations of wearout mechanisms are discussed. The conclusions are drawn in Section IV.

II. Inverse Design Based on Deep Neural Networks & an Evolutionary Algorithm and the Wearout Mechanisms

A. Wearout Mechanisms and Process Parameters

Fig. 1 shows an example of a 6T SRAM cell to be studied. There is variation in parameters due to both manufacturing process and wearout. The time-zero variability is assumed to be due to gate length (described in Table I) and the threshold voltages shift of the six devices in the cell.

978-1-7281-3199-3/20/$31.00 ©2020 IEEE

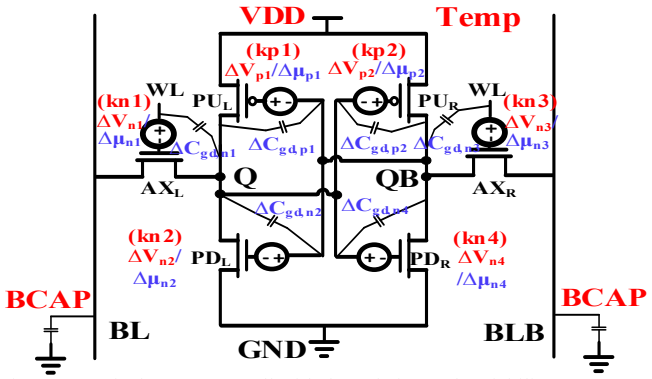


Fig. 1. A typical 6T SRAM cell with degradation and variability parameters marked.

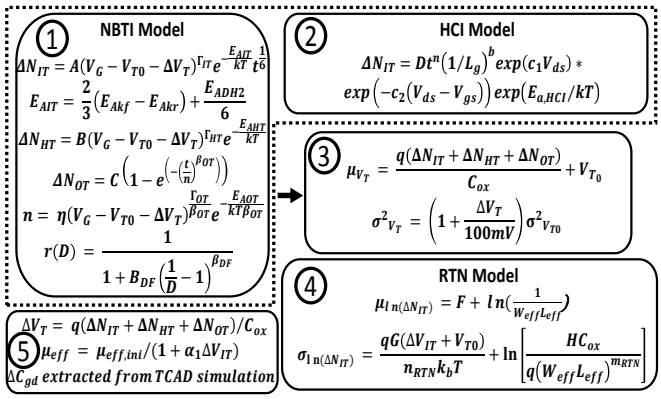


Fig. 2. The analytical expressions for NBTI, HCI, and RTN [5]-[11].

In this study, we assume that NBTI, HCI, and RTN cause the wearout of the FinFET SRAM cell. According to the description in [5], NBTI is modelled with three uncorrelated parts, which are interface traps (ΔN_{IT}) and the pre-existing (ΔN_{HT}) and generated (ΔN_{OT}) traps in the bulk. These parameters are a function of stress time and recovery time. The time range and the ratios of stress and recovery are important for suitable prediction for NBTI. A complete solution of NBTI under long-term DC stress is given in [6]. We combine the long-term DC stress model with a duty factor ratio equation (for recovery) to calculate the overall NBTI degradation. A simplified expression of the recovery fraction in [6] is adopted for NBTI prediction. With this method, we incorporate the effect of stress and recovery cycles using the duty ratio (marked as D in Fig. 2). The duty ratio highly depends on the data stored in SRAM cells. Since we study 6T FinFET SRAM cells which have symmetric structure, a duty ratio closer to 0 or 1.0 leads to more degradation than a duty ratio of 0.5.

HCI is modeled with a shift in interface charge (ΔN_{IT}) and is considered be not recoverable [9]. Since HCI happens when a transistor is on and is conducting current, HCI is determined by the transition rate (TR) of data stored in cells. An equivalent time gap is applied to simulate the transition of each data flip in an SRAM cell. The effective stress time related to HCI (marked as t in box No. 2 of Fig. 2) is the product of overall stress time, the equivalent time gap for each transition, and TR. The interface trap shift due to HCI varies with effective stress time (t), device dimensions, stress voltage, and temperature [7], [8].

Random process parameters cause time-zero variability in threshold voltage. The overall shift of the threshold voltage due to NBTI and HCI is modeled as a normal distribution [19] and is added to the time-zero variability in threshold voltage. RTN introduces extra variation in the interface trap density. However, RTN doesn't affect the accumulated ΔN_{IT} induced by either NBTI or HCI because its impact is considered be temporary. The impact of RTN is modelled as an independent lognormal distribution. For each sample, the deviation of the ΔN_{IT} from the lognormal distribution due to RTN is added to the deviation of ΔN_{IT} from the normal distribution due to NBTI and HCI.

There are various models which are proposed to describe theses wearout mechanisms. It is necessary to include the impact on all device parameters for this study. In general, it's assumed that NBTI and HCI bring about a shift of charge density in transistors which leads to shifts of threshold voltage (ΔV_T), carrier mobility (μ), subthreshold slope (SS), and the gate-drain capacitance (ΔC_{gd}). RTN intensifies the impact of NBTI and HCI with extra variation. Fig. 2 shows the analytical expressions for NBTI and HCI in box No.1 and box No.2. The threshold voltage distribution due to NBTI, HCI, and time-zero variability are listed in box No.3. The extra deviation induced by RTN is described in box No. 4. Since box No.1 and box No.2 just describe how the trap density shifts with stress time, the relationship between the shift of trap density and shift of threshold voltage (ΔV_T) and the relationship between ΔV_T and effective carrier mobility (μ_{eff}) are given in box No. 5. Box No.5 also describes that the deviation of the gate-drain capacitance due to wearout is extracted with TCAD simulations. It's found that the SS is insensitive to wearout, and therefore it's not included in our analysis. Since μ and ΔC_{gd} can be expressed as direct functions of the threshold voltage deviation, they are not input variables for the DNNs in the next part. Their value has been updated as a function of ΔV_T when the dataset is generated with SPICE simulations.

The parameters related to wearout are marked in Fig. 1. They are the threshold voltage shifts (ΔVn1, ΔVn2, ΔVn3, ΔVn4, ΔVp1, ΔVp2) and their ratios due to interface traps for each transistor (kn1 = $\Delta Vn1_{\_IT}/\Delta Vn1$, kn2 = $\Delta Vn2_{\_IT}/\Delta Vn2$, kn3 = $\Delta Vn3_{\_IT}/\Delta Vn3$, kn4 = $\Delta Vn4_{\_IT}/\Delta Vn4$, kp1 = $\Delta Vp1_{\_IT}/\Delta Vp1$, and kp2 = $\Delta Vp2_{\_IT}/\Delta Vp2$). Variability due to wearout is assumed to be caused by ΔVn1 to ΔVp2.

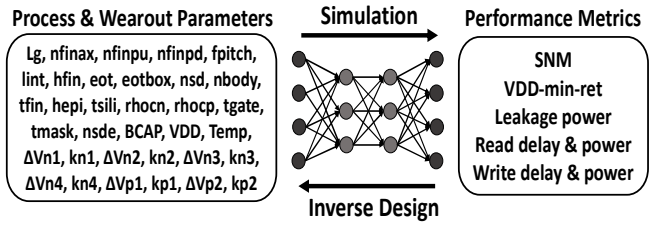
This study considers both process and wearout parameters. The relevant process parameters are listed in Table I. These parameters, except for Lg, are optimized in the presence of variation in the process and due to wearout. Their value is also updated in device model files when generating the dataset. Lg isn't chosen as one of the optimization parameters because its variation leads to a shift in the wearout parameters.

B. *Inverse Design*

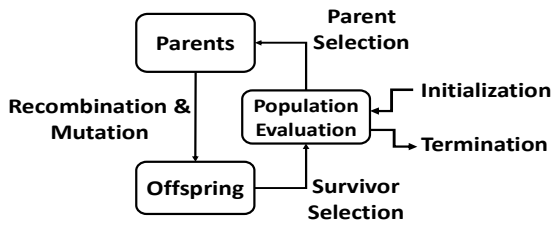
Fig. 3 shows the architecture of the proposed method for inverse design, and the general flowchart of the evolutionary algorithm. The relationships between cell parameters (related to the process and wearout) and performance metrics (we considered SNM, VDD-ret-min, and read delay & power as examples) are built with deep neural networks. All the input and output variables of the DNNs are normalized for convenient and quick training. Then the cell's various process parameters are determined through the evolutionary algorithm to meet requirements on the performance metrics. The impact of wearout mechanisms can be incorporated as well. It should be kept in mind that DNNs with high accuracy for the relationship between process and wearout parameters and performance metrics ensures the final results of our inverse design.

Table I. Description of process parameters [20]

Lg	Gate length	nfinax	Finger number of access transistor
nfinpu	Finger number of pull up transistor	nfinpd	Finger number of pull down transistor
fpitch	Fin pitch	lint	Length reduction parameter
hfin	Fin height	eot	SiO_2 equivalent gate dielectric thickness
eotbox	SiO_2 equivalent buried oxide thickness	nsd	S/D doping concentration
nbody	Channel (body) doping concentration	tfin	Body (fin) thickness
hepi	Height of the raised source/drain on fin top	tsili	Thickness of silicide on top of raised source/drain
rhocn	Contact resistivity at the silicon/silicide interface of nFET	rhocp	Contact resistivity at the silicon/silicide interface of pFET
tgate	Gate height on top of the hard mask	tmask	Height of the hard mask on top of the fin
nsde	Active doping concentration at the channel edge		



(a)



(b)

Fig. 3. (a) Architecture of the proposed method for inverse design, and (b) the general flowchart of the evolutionary algorithm.

Fig. 3(a) shows input and output variables of DNN. These input variables of the DNNs include wearout parameters, process parameters, and environmental parameters, such as the supply voltage and temperature. The process parameters listed in Table I need to be optimized. Others parameters considered include VDD and Temp, which relate to the operating conditions of the circuit, and also include shifts in wearout parameters, including the random variation in these shifts.

The evolutionary algorithm (EA) that is used for optimization is adopted from biological evolution in nature. There are many different dialects for the EA. It is called the Genetic Algorithm (GA) when the candidates are strings, Genetic Programming (GP) when the candidates are finite state machines, and Evolution Strategies (ES) when the candidates are real-valued vectors, etc. [21], [22]. However, the basic principle for these EAs is almost the same. Because of the existence of natural selection, the candidates of the first generation for specific applications are evaluated with an estimator which measures how good they are in a natural environment. Next, the second generation of candidates are randomly created on the basis of good candidates from the first generation. After that, the above mentioned two steps are repeatedly implemented to obtain new generations until the final generations are obtained which can meet the requirements with an acceptable confidence.

In our study, each candidate is a vector which contains process parameters and device parameters' shift due to wearout of the FinFET SRAM cell. Our estimation process checks how performance metrics meets our expectations. In the initial state, we generate a bunch of candidates which contain randomly adopted process and wearout parameters.

Then some of these candidates are selected to be parents of the next generation based on how well they match design requirements (minimum/maximum limits) on the performance metrics. After that, re-combinations and mutations are applied on parents to reproduce their offspring. A portion of offspring which survive in the environment related to the design target forms the new generation. Afterwards, the most excellent individuals of the new generation are taken as the parents of the next round of evolution.

C. The Application of Deep Neural Networks

The structure of deep neural networks is shown in Fig. 3(a). It consists of an input layer, an output layer, and several hidden layers which are used to emulate the relationship between the input layer variables and the output layer variables.

The input variables include all the process parameters in Table I, together with the time-zero and wearout parameters mentioned in Section 2B. In addition, we also include the BL/BLB (bit-line, bit-line bar) capacitance (BCAP), temperature (Temp), and power supply voltage (VDD) as input variables of the DNNs to ensure comprehensiveness of the predicted results. Note that VDD is not an input variable of the DNN for VDD-min-ret because it is swept to obtain VDD-min-ret during the phase of data generation.

Since we show the inverse design for SNM, VDD-min-ret, and read delay & power as examples, three DNNs for each type of performance metric are trained separately. Another reason for separate training is that the goals on these performance metrics don't have an obvious connection.

The training and test datasets are obtained from SPICE simulations. For each sample, the randomly adopted input variables are embedded into the netlist. Then the trial-error simulations are executed to get SNM and VDD-min-ret. And the read delay & power are extracted directly. In the dataset, there are 15k samples in total for each performance metric. The sample ratio for training and test is 9:1. Each of the DNNs has four hidden layers with 128 neurons for each layer. The DNN for SNM has 34 input variables and one output variable, the DNN for VDD-min-ret has 33 input variables and one output variable, while the DNN for read delay & power has 34 input variables and four output variables because read operations on '0' and '1' have different delays and power. The training and test procedures are implemented with PyTorch [23]. The training procedure is based on batch gradient descent with an L2 (least square errors) function and an Adam optimizer. In the first step, the model is trained for 10k epochs with a learning rate of 0.001. Then the model is trained with a learning rate of 0.0001. In general, the overall test accuracy gets higher than 99% after being trained for another several thousand epochs. During the training process, the test accuracy is monitored carefully to prevent overfitting. It's noted that the DNN structure and training strategies can be adjusted to ensure prediction accuracy.

Fig. 4 shows the comparison of normalized labels with output variables predicted by DNNs for several performance metrics. Clearly, the DNNs can predict performance metrics under various cell configurations with good accuracy. Based on the DNNs, the exploration of cell configurations for each performance metric is implemented with an EA as described in Section II.

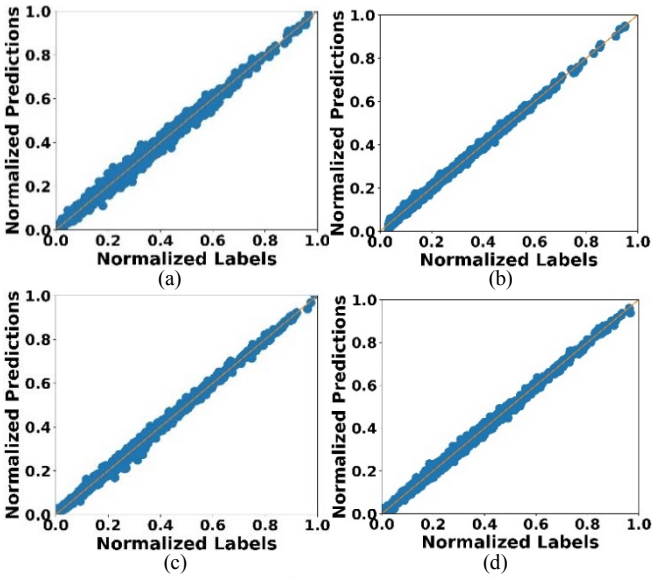


Fig. 4. The comparison of normalized labels with normalized predictions by the DNNs for (a) SNM, (b) VDD-min-ret, (c) delay of read '0', and (d) power of read '0'. The comparison for delay and power of read '1' is similar.

III. Application of Inverse Design on FinFET SRAM Cell

In this section, we study the application of inverse design to FinFET SRAM cell performance metrics. We specifically have considered SNM, VDD-min-ret, and read delay & power. The inverse design for other/additional performance metrics can be applied with the same method. As mentioned in Section II, the DNNs for relationships between cell parameters and performance metrics (SNM, VDD-ret-min, and read delay & power) are obtained in the first step. Then our evolutionary algorithm is applied to find the process parameters which ensure the target performance metrics meet our expectations. The case with process parameter variations in the fresh state, the case with variations related to process and wearout mechanisms after specific stress times, and the case with variations related to process and wearout by subset of wearout mechanisms are studied.

A. Cases with Process Parameters Variations

In this part, the inverse design is explored for a FinFET SRAM cell in the fresh state while considering process parameter variations. Since the requirements on different performance metrics can't always be achieved at the same time, the explorations of the SNM, VDD-min-ret, read delay & power, and other performance metrics are implemented separately. Read/Write delay and power are different for operations on '0' and '1'.

In this work, we show the results for SNM, VDD-min-ret, and read delay as examples. Obviously, the input variables of Temp and VDD can't be adjusted arbitrarily, and we set them as constants of 0.685 and 1.0 after normalization, in the following inverse designs. The mean value of Lg is set as 0.5 after normalization. For the cases shown in Fig. 5, the normalized specifications for SNM, VDD-mint-ret, and read 0 delay are 0.3, 0.5, and 0.5, respectively.

Since a larger SNM is preferred and the process variations introduce more uncertainty, our algorithm is designed to find the process parameters for the FinFET SRAM cell (as shown in Fig. 5(b)) to ensure that the worst case (smallest) SNM meets the required specification. The blue line is a reference for a good Weibull distribution.

In contrast, smaller VDD-min-ret and read 0 delay are preferred. The algorithm helps obtain deterministic values of process parameters to ensure that even the largest VDD-min-ret and read delay meet the specifications under process variations, with the process parameters listed in Figs. 5(d) and 5(f). It's noted that the value of some process parameters (such as nfinax, nfinpu, and nfinpd) after inverse design is 0. This means that the absolute number of fingers is 1 instead of 0, because we have normalized the number of fingers with a smallest value of 1 and a largest value of 10. This is similar for the other process parameters. Since the EA is implemented with a random selection of parameters, the process parameters are different after each inverse design step although the targets are kept the same.

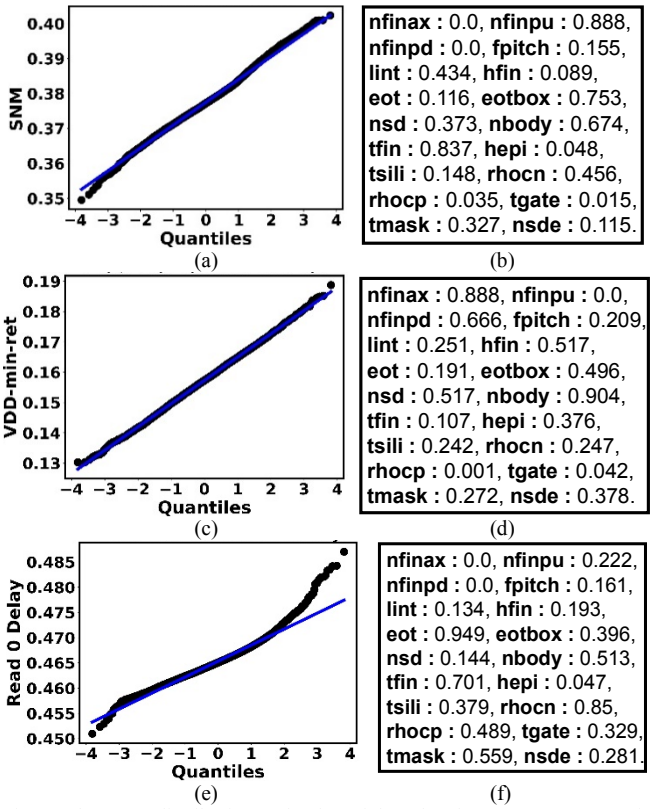


Fig. 5. The SNM distribution and value of the related process parameters in (a) and (b), the VDD-min-ret distribution and value of the related process parameters in (c) and (d), and the read 0 delay distribution and value of the related process parameters in (e) and (f), after inverse designs when the circuit is in fresh state.

Although the process parameters are randomly picked in the EA to emulate the mutation procedure, it's necessary to check how important each process parameter is for the performance metrics, to understand the evolution of the process parameter set. We adopt a strategy called random permutation to achieve this goal. It consists of 18 steps (because we have 18 process parameters) for each performance metric. At each step, the importance of one process parameter X_i for a specific performance metric Y_j is studied (i ranges from 1 to 18, j ranges from 1 to 3). A Monte Carlo simulation with a specified sample size (1000) is applied to each step. The process parameters, except for a specific X_i, are randomly selected between 0.0 and 1.0 for each sample. Then the increase of the performance metric (ΔY_j) due to the increase of X_i from 0.0 to 1.0 is recorded for each sample. At last, the averaged increase of the performance metric is used to measure the importance of X_i to Y_j. It's observed that the trend of Y_j due to X_i is monotonous. Fig. 6 shows the most important process parameters for SNM, VDD-min-ret, and read 0 delay in the fresh state. The process

parameters corresponding to the averaged ΔY_j with an absolute value that is smaller than 0.05 are considered be unimportant, and thus are not shown. The red arrow pointing up denotes that an increase of X_i brings about a larger Y_j, and vice versa. According to the description in Fig. 6, a larger nfinax, eot, and tfin, and a smaller nfinpd and fpitch can help improve SNM. A smaller nfinax, and a larger nfinpu and eot brings about a better (smaller) VDD-min-ret. Read 0 delay is affected by various process parameters, in which nfinax, nfinpd, and hfin are the most crucial. It's noted that the EA doesn't necessarily make all parameters evolve in the directions which are beneficial for the target on the performance metric if the evolution directions of some process parameters don't disturb the overall exploration for the design target. For example, it's acceptable to keep nfinax as 0 (shown in Fig. 5 (b)) because the nfinpd of 0.0 is good enough to ensure all samples of the SNM distribution stay on the right-hand side of the target.

nfinax -> ↑0.301, **nfinpd** -> ↓0.367, **fpitch** -> ↓0.052, **eot** -> ↑0.111, **tfin** -> ↑0.095.

(a)

nfinax -> ↑0.062, **nfinpu** -> ↓0.127, **eot** -> ↓0.229.

(b)

nfinax -> ↓0.169, **nfinpu** -> ↓0.112, **nfinpd** -> ↓0.415, **fpitch** -> ↓0.139, **hfin** -> ↓0.267, **eot** -> ↑0.056, **nbody** -> ↑0.054, **tfin** -> ↓0.137, **rhocn** -> ↑0.145.

(c)

Fig. 6. The most important process parameters for (a) SNM, (b) VDD-min-ret, and (c) read 0 delay, for the circuit in fresh state.

nfinax -> ↑0.456, **nfinpu** -> ↓0.061, **nfinpd** -> ↓0.521, **fpitch** -> ↓0.094, **hfin** -> ↑0.056, **eot** -> ↑0.092, **nbody** -> ↓0.078.

(a)

nfinax -> ↑0.055, **nfinpu** -> ↓0.122, **eot** -> ↑0.093.

(b)

nfinax -> ↑0.096, **nfinpu** -> ↓0.412, **nfinpd** -> ↓0.942, **fpitch** -> ↓0.404, **lint** -> ↓0.116, **hfin** -> ↓0.561, **eot** -> ↑0.335, **eotbox** -> ↓0.052, **nsd** -> ↑0.174, **nbody** -> ↑0.091, **tfin** -> ↓0.059, **tsili** -> ↓0.115, **rhocn** -> ↑0.365.

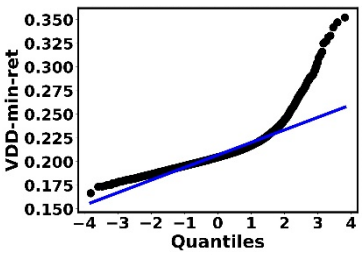
(c)

Fig. 7. The most important process parameters for (a) SNM, (b) VDD-min-ret, and (c) read 0 delay, when the circuit has been under 10 years of stress (duty cycle of 0.1, and transition rate of 20 times/µs).

B. Cases with Variations Related to Wearout Mechanisms and Cases with Variations Related to Fewer Wearout Mechanisms

The most important parameters for each performance metric have been analyzed for a circuit in the fresh state in part A. Since wearout mechanisms impact device parameters, the relevant importance of these parameters for performance metrics get be different when wearout is taken into consideration. Fig. 7 shows the most important process parameters for SNM, VDD-min-ret, and read 0 delay, when the circuit has been has received 10 years of stress. Based on the comparison between Fig. 6 and Fig 7, it is observed that the wearout mechanisms make SNM and read 0 delay sensitive to more parameters. For example, SNM becomes sensitive to nfinpu, hfin, and nbody, and is more sensitive to nfinax and nfinpd. Read 0 delay becomes sensitive to lint, eotbox, nsd, and tsili, and becomes less sensitive to nfinax and tfin. Inverse design is also implemented for SNM, VDD-min-ret, and read 0 delay with the impact of NBTI, HCI, and RTN. The specifications on them are still 0.3, 0.5, and 0.5, respectively. The results of the inverse design for SNM and read 0 delay are similar with the results in part A. However, as shown in Fig. 8, the final distribution of VDD-min-ret is always much more non-Weibull than the distribution in part A, which is mainly due to the occurrence of RTN. The procedures for each inverse design step are accomplished in several seconds.

Moreover, RTN leads to the failure of inverse design for certain cases. A good example is the inverse design for SNM. According to the description in Fig. 6 and Fig. 7, a larger nfinax, hfin, and eot are crucial for getting a larger SNM. Inverse design fails if we set nfinax, hfin, and eot as 0 with a design target of 0.7 for circuits which have been stressed by NBTI, HCI, RTN for 10 years, while the designs with the same settings in the fresh state or under stress by only NBTI and HCI can be optimized. Fig. 9 shows the SNM distributions after inverse design for a circuit in the fresh state and when the circuit has been stressed for 10 years.

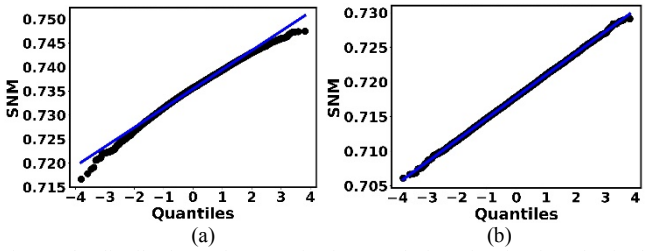


(a)

nfinax : 0.0, **nfinpu :** 0.222, **nfinpd :** 0.0, **fpitch :** 0.161, **lint :** 0.134, **hfin :** 0.193, **eot :** 0.949, **eotbox :** 0.396, **nsd :** 0.144, **nbody :** 0.513, **tfin :** 0.701, **hepi :** 0.047, **tsili :** 0.379, **rhocn :** 0.85, **rhocp :** 0.489, **tgate :** 0.329, **tmask :** 0.559, **nsde :** 0.281.

(b)

Fig. 8. The VDD-min-ret distribution and value of the process parameters in (a) and (b), after inverse designs, when the circuit has been under 10 years of stress (duty cycle of 0.1, and transition rate of 20 times/µs).



(a) (b)

Fig. 9. The distributions of SNM, after inverse designs for circuit under fresh state (a), and under the state when the circuit has been under 10 years of stress (b). The duty cycle is 0.1, and the transition rate is 20 times/µs. Only the impact of NBTI and HCI is included.

IV. Conclusion

We have proposed a methodology based on DNNs and an EA to do inverse design for FinFET SRAM cells. Each step of the overall inverse design can be accomplished as fast as several seconds. This inverse design technique provides convenience and high efficiency for designers. No matter whether the cell is in the fresh state or is stressed by various wearout mechanisms, the algorithm can help the user obtain configurations (process parameters) which make the cells' performance metrics meet specific targets, although various variations exist. The EA doesn't make all parameters evolve in the directions which are beneficial for target on performance metric if the evolution directions of some process parameters don't disturb the overall exploration for target.

Moreover, RTN leads to the failure of inverse design for certain cases.

ACKNOWLEDGEMENT

The authors would like to thank the NSF for support under Award Number 1700914.

REFERENCES

[1] P. M. Solomon, K. W. Guarini, Y. Zhang, K. K. Chan, E. C. Jones, G. M. Cohen, A. Krasnoperova, M. Ronay, O. Dokumaci, H. J. Hovel, J. J. Bucchignano, C. Cabral Jr., C. Lavoie, V. Ku, D. C. Boyd, K. S. Petrarca, J. H. Yoon, I. V. Babich, J. Treichler, P. M. Kozlowski, J. S. Newbury, C. P. D'Emic, R. M. Sicina, J. Benedict, and H. -S. P. Wong, "Two gates are better than one," *IEEE Circuits and Devices Magazine*, vol. 19, no. 1, pp. 48-62, 2003.

[2] E. J. Nowak, I. Aller, T. Ludwig, K. Kim, R. V. Hoshi, C. -T. Chuang, K. Bernstein, and R. Puri, "Turning silicon on its edge [double gate CMOS/FinFET technology]," *IEEE Circuits and Devices Magazine*, vol. 20, no. 1, pp. 20-31, 2004.

[3] T. Skotnicki, J. A. Hutchby, T. -J. King, H. -S. P. Wong, and F. Boeuf, "The end of CMOS scaling: toward the introduction of new materials and structural changes to improve MOSFET performance," *IEEE Circuits and Devices Magazine*, vol. 21, no.1, pp. 16-26, 2005.

[4] "International technology roadmap for semiconductors,": http://www.itrs2.net/2011-itrs.html.

[5] K. Joshi, S. Mukhopadhyay, N. Goel, and S. Mahapatra, "A consistent physical framework for N and P BTI in HKMG MOSFETs", in *Proc. IEEE Int. Reliability Physics Symp.*, no. 6241840, pp. 5A.3.1-5A.3.10 , 2012.

[6] A. Herrera-Moreno, J. L. García-Gervacio, H. Villacorta-Minaya, and H. Vázquez-Leal. "TCAD analysis and modeling for NBTI mechanism in FinFET transistors," *IEICE Electronics Express*, vol. 15, no. 14, pp. 1-12, 2018.

[7] I. Messaris, T. A. Karatsori, N. Fasarakis, C. G. Theodorou, S. Nikolaidis, G. Ghibaudo, and C. A. Dimitriadis, "Hot carrier degradation modelling of short-channel n-FinFETs suitable for circuit simulators," *Microelectron. Reliab.*, vol. 56, pp. 10-16, 2016.

[8] E. Chung, K. Nam, T. Nakanishi, S. Park, H. Yang, T. Kauerauf, G. Jiao, D. Kim, and K. H. Hwang, "Investigation of hot carrier degradation in bulk FinFET," in *Proc. IEEE Int. Reliability Physics Symp.*, XT-6.1- XT-6.4, 2017.

[9] C. D. Young, J.-W. Yang, K. Matthews, S. Suthram, M. M. Hussain, G. Bersuker, C. Smith, R. Harris, R. Choi, B. H. Lee, and H-H. Tseng, "Hot carrier degradation in HfSiON/TiN fin shaped field effect transistor with different substrate orientations," *J. Vac. Sci. Technol. B*, vol. 27, no. 1, pp. 468-471, 2009.

[10] N. Tega, H. Miki, F. Pagette, D. J. Frank, A. Ray, M. J. Rooks, W. Haensch, and K. Torii, "Increasing threshold voltage variation due to random telegraph noise in FETs as gate lengths scale to 20 nm," in *Symp. VLSI Technol. Dig.*, pp. 50-51, 2009.

[11] K. Sonoda, K. Ishikawa, T. Eimori, and O. Tsuchiya, "Discrete dopant effects on statistical variation of random telegraph signal magnitude," *IEEE Trans. Electron Devices*, vol. 54, no. 8, pp. 1918-1925, 2007.

[12] K. X. Yang, T. Z. Liu, R. Zhang, and L. Milor. "A Comparison Study of Time-Dependet Dielectric Breakdown for Analog and Digital Circuit's Optimal Accelerated Test Regions," in *Proc. Design of Circuits and Integrated Systems Conf.*, 2017.

[13] R. Zhang, T. Z. Liu, K. X. Yang, and L. Milor, "Modeling of the Reliability Degradation of a FinFET-based SRAM Due to Bias Temperature Instability, Hot Carrier Injection, and Gate Oxide Breakdown," in *Proc. Int. Integrated Reliability Workshop*, 2017.

[14] R. Zhang, T. Z. Liu, K. X. Yang, and L. Milor, "Modeling for SRAM reliability degradation due to gate oxide breakdown with a compact current model," in *Proc. Design of Circuits and Integrated Systems*, 2017.

[15] K. X. Yang, T. Z. Liu, R. Zhang, and L. Milor, "A Novel Front-End-of-Line (FEOL) and Middle-of-Line (MOL) Time-Dependent Dielectric Breakdown Lifetime Simulator for FinFET Technology," in *Proc. IEEE VLSI Test Symp.*, 2018.

[16] Z. Liu, D. Zhu, S. P. Rodrigues, K. -T. Lee, W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Lett.*, vol. 18, no. 10, pp. 6570-6576, 2018.

[17] H. Yu, J. Shin, T. Michalka, M. Larbi, and M. Swaminathan, "Behavioral modeling of tunable I/O drivers with pre-emphasis using neural networks," in *Proc. 20th Int. Symp. Qual. Electron. Design* (*ISQED*), pp. 247-252, 2019.

[18] H. Yu, H. Chalamalasetty, and M. Swaminathan, "Modeling of voltagecontrolled oscillators including I/O behavior using augmented neural networks," *IEEE Access*, vol. 7, pp. 38973-38982, 2019.

[19] P. Weckx, B. Kaczer, M. Toledano-Luque, P. Raghavan, J. Franco, P. J. Roussel, G. Groeseneken, and F. Catthoor, "Implications of BTI-induced time-dependent statistics on yield estimation of digital circuits," *IEEE Trans. Electron Devices*, vol. 61, no. 3, pp. 666-673, 2014.

[20] V. Sriramkumar, N. Paydavosi, D. Lu, C. -H. Lin, M. Dunga, S. Yao, T. Morshed, A. Niknejad, and C. Hu, "BSIM-CMG106.0.0 Multi-Gate MOSFET Compact Model," *Technical Manual*, 2012.

[21] C. A. Coello Coello, "An Introduction to Evolutionary Algorithms and Their applications," in Lecture Notes in Computer Science, Springer Berlin/Heidelberg, vol. 3563, pp. 425-442, 2005.

[22] A. N. Sloss and S. Gustafson, "2019 Evolutionary Algorithms Review," arXiv preprint arXiv:1906.08870, 2019.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Neural Information Processing Systems*, 2017.