Bridge-Defect Prediction in SRAM Circuits Using Random Forest, XGBoost, and LightGBM Learners

Joydeep Ghosh
Electrical & Computer Engineering
National University of Singapore
Singapore
elejg@nus.edu.sg

Shang Yi Lim
Electrical & Computer Engineering
National University of Singapore
Singapore
e0529483@u.nus.edu

Aaron Voon-Yew Thean
Electrical & Computer Engineering
National University of Singapore
Singapore
aaron.thean@nus.edu.sg

***Abstract*—The modern fabrication technology node for devices and circuits is causing several failure analysis challenges of the current state-of-the-art tools to actually find the location of the physical defects. Here, we show an approach of combining SPICE circuit simulations and machine learning for bridging-detect predictions in a 6T-SRAM cell. Our predictive model consists of Random Forest, XGBoost, and LightGBM algorithms which are trained with several electrical attributes of the circuit having faults. We compare the performance of the algorithms and calculate the accuracy in predicting the defect location. We believe that this approach promises to improve the failure analysis, enhancing the cycle of design to product.**

Keywords—Failure analysis, 6T-SRAM, Random Forest, XGBoost, LightGBM

I. INTRODUCTION

A continuous reduction of the technology node is the primary reason behind the amazing increase of the computational power of modern integrated circuits (ICs). The increasing packing density is allowing more logic circuits to be fabricated on a given area of the IC chip. The transistor structure in the circuits has drastically changed from a planar architecture to a 3D fin field-effect transistors (FinFETs) [1] and then gate-all-around (GAA) nanosheets [2]. New materials and technology apart from conventional silicon have also been introduced [3, 4]. To support all these, the process technology is becoming more and more complex.

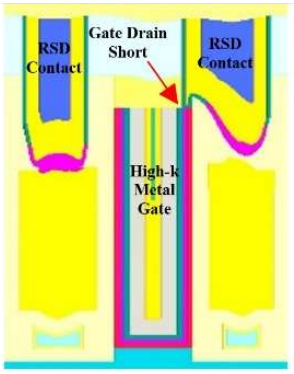

Fig.1: Simulated gate-drain short between contact and high-k metal gate in a nMOS FinFET by Sentaurus SPX is shown.

Failure Analysis (FA) is a major way to drive yield enhancement, reliability, and accelerate product development cycle in the semiconductor industry. The continuous miniaturization of the semiconductor devices has led to the dimension reduction of circuits (e.g. a 6T-SRAM cell) yet achieving improved performances. Because SRAM-based systems are designed with a tight design rules, one expects them to provide an appropriate vehicle to diagnose failures [5, 6]. With modern technology nodes, the defect location identification without any prior knowledge of any electrical characteristics data is almost an impossible task. Among other defects, a resistive short (or bridging-defect) is often considered as a common type of yield killer particularly for front-end of line (FEOL) yield. A slight variation during the processing can result in shorts of local interconnect as well as any of a transistor in a 6T-SRAM bit cell [7, 8].

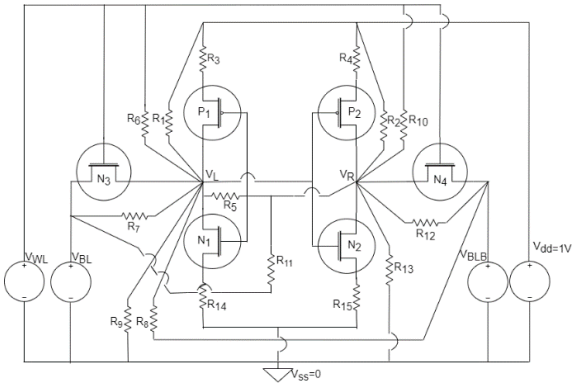

Fig.2: A 6T-SRAM bit cell with bridge-defects as resistors in between different possible node-pairs is shown. In this work, one defect at a time is assumed to be present. Industry-standard BSIM-CMG compact model is used for simulations. The resistors R_1 to R_{15} are varied from a lower to higher value to create sufficient data for algorithm training. BL and BLB- two bit lines, WL- word line.

The defect identification workflows are dependent on the electrical nano-probing method. This nano-probing is a SEM (scanning electron microscopy)-based probing technique collecting the DC characteristics for analysis of the transistor-level behavior. The time and the cost required to perform such physical FA using industry-standard FA tools is generally high, and the electrical interactions between the defect, transistors, and interconnects are becoming difficult to partition. Moreover, as the occurrence chance of real defects is quite low in microchips, a defect modeling and circuit simulation methodology should be adopted.

II. SIMULATIONS

Fig. 1 shows an example of a nMOS device in the presence of such a resistive short defect, generated by simulations through Synopsys Sentaurus Process Explorer (SPX). SPX is a new physical layout-based TCAD structural modeler, used for visualization and understanding of the defect origination [9]. SPX provides a graphical user interface (GUI) to create layouts and device process flow. It provides a Design of

978-1-6654-0685-7/21/$31.00 ©2021 IEEE

Experiment (DOE) interface for simulating multiple structures with varying process parameters for comparison. The TCAD tool SPX, as compared to another popular TCAD tool Sentaurus Process (SProcess), is a more efficient choice to simulate complex 3D structures such as a FinFET device for its own reasons. The created device structure can be further used for device (as well as mixed-mode) simulations using the Sentaurus Device (SDevice) tool. The major characteristics of the SPX tool over SProcess is the complex mesh system demanding special treatment explained in [10]. Indeed, one can use TCAD-based simulations to understand the impact of such defects on a 6T-SRAM circuit. However, this approach will involve a complex mesh system of the SPX as well as much higher simulation time with SDevice.

In this work, with SPICE-level simulations [11] we scrutinize, if one can use advanced data analytics or machine learning (ML) technique for failure mode identification. In fact, different ML techniques in the last few years have already been utilized in the research domain to understand several semiconductor device-level failures, both in the presence and absence of complete domain knowledge [12-16].

Fig. 2 shows a SRAM circuit having a bridge-defect present across any one of the fifteen possible node pairs. Each such resistor may refer to either interconnect-level or device-level short, e.g. the bridge-resistor R_1 (R_2) represents source-drain short to P_1 (P_2) transistor. Our first step is to understand how the presence of such a resistance impacts the output characteristics of the circuit. We have to identify the circuit features which are easily measurable and influenced by the defect locations. For ML training purpose, we have to develop a dataset comprising of such features. Then, during the testing time, the algorithm should predict the defect location based on the electrical attributes provided. The simulation methodology is described in the following section.

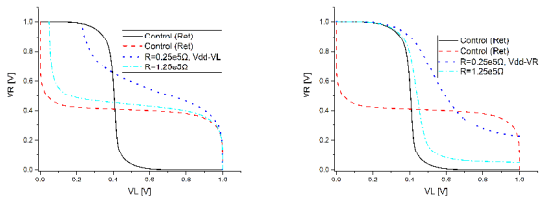

Fig.3: The SRAM butterfly curves during the retention mode of operation (i.e. the circuit biased at $V_{WL}=V_{BL}=V_{BLB}=V_{ss}=0$, $V_{dd}=V_{DD}=1V$) having the defect resistance R present at two different locations (i.e. R_1 referring to left and R_2 referring to right figure) is shown, c.f. Fig. 2. The curve for the control circuit (i.e. no defect) is also shown for a comparison. Clearly, the presence of the defects at different locations result in different values of static noise margin and the point where $V_L=V_R$.

A. *Spice Simulations*

Industry-standard BSIM-CMG multi-gate compact model with default parameters is chosen for SRAM circuit simulations. We extract the features of the circuit from both of its DC and transient analyses in the presence of a resistor across each of the node pairs. We vary the resistance values through simulations to create enough data. Of course, the value of the resistance will be proportional to the size of the defect. The static noise margin (SNM), an important parameter pertaining to the stability of the bit cell, is obtained by adjusting the largest possible square in between the two voltage transfer curves (VTC, forming a butterfly-like curve) of two CMOS inverters. In our case, the DC features are extracted from the so-called 'butterfly curve' during the conventional retention (c.f. Fig. 3, the voltage bias is given by $V_{WL}=V_{BL}=V_{BLB}=V_{ss}=0$, $V_{dd}=1V$) and read (voltage bias is $V_{WL}=V_{BL}=V_{BLB}=V_{dd}=1V$, $V_{ss}=0V$) modes of operations. For example, the trip point voltage, defined as the point where the line $V_R=V_L$ meets the VTC curve of the respective inverter is considered as a feature. From the symmetry of the circuit we understand that one has to vary V_R in order to locate any defect present in between any of V_{dd}/BL/BLB/WL/V_{ss} points and V_L. Similarly, we should vary V_L to find defects present in between any of the V_{dd}/BL/BLB/WL/V_{ss} points and V_R. Again, the coordinates of the points where slope of the VTC curves become -1 are chosen as features.

In Fig. 3(left) when V_R is drawn to a high value, ideally P_1 would be open circuited and N_1 would be conductive. The presence of the resistance R_1 acts as the source-drain short of P_1, thus a current would flow along V_{dd}->R_1-> N_1 direction, causing a voltage drop across N_1. This explains the corresponding VTC curve. In similar, as noticed in Fig. 3(right) when V_L is drawn to a high value, P_2 should ideally be open and N_2 conductive. The presence of the resistance R_2 would act as the source-drain short of P_2, thus a current would flow along V_{dd}->R_2->N_2, causing a voltage drop across N_2. This explains the corresponding VTC curve. It is also noted that when the resistance value increases, both the 'butterfly' curves tend towards the one without any defect. This is obvious as in such a case, the shorting resistance is becoming open circuited.

During the transient analysis we write a logic such that the state of V_R (V_L) switches to 'low' ('high') from initial 'high' ('low') value. This means, we pre-charge one of the bit lines to high and the other to low, then turn on the access transistors using the word line pulse to change their states. By this way, we consider the rise time (change from 10% to 90%) of V_L, fall time (change from 90% to 10%) of V_R, and the point coordinate where those two signals meet (i.e. $V_L(t)=V_R(t)$) as features. By all these means, we have successfully extracted more than twenty features from DC and transient simulations.

Testing (%)	20	30	40	50	60	70
RF	96.67	97.04	97.22	96.89	97.41	89.52
XGBoost	94.44	97.04	96.67	96.44	95.55	90.47
LightGBM	94.44	95.55	96.11	96.88	97.04	93.65

Table I: The table shows the prediction accuracy (%) of the chosen algorithms, where training data is gradually decreased. Here, we have used preprocessing technique like 'StandardScaler' from 'sklearn' [21].

B. *Machine Learning Implementation*

Generally, the supervised ML refers to using a set of input features to predict the value of a labeled output variable (i.e. defect location in our case). Once the database is developed, three supervised learning algorithms e.g. Random Forest (RF) [17], Extreme Gradient Boosting (XGBoost) [18], and LightGBM [19] are chosen for training and predicting the defect location based on the input attributes. The main advantage of these algorithms is their ability to extract important features from the trained predictive model. Generally, the execution time of XGBoost is lower than that of LightGBM. We mention here that for certain values of the resistance in our chosen range, our dataset would contain some missing values. In order to circumvent this issue, when necessary, we have used the imputation technique for filling the missing values using k-Nearest Neighbors ('KNNImputer') algorithm from 'sklearn' [20].

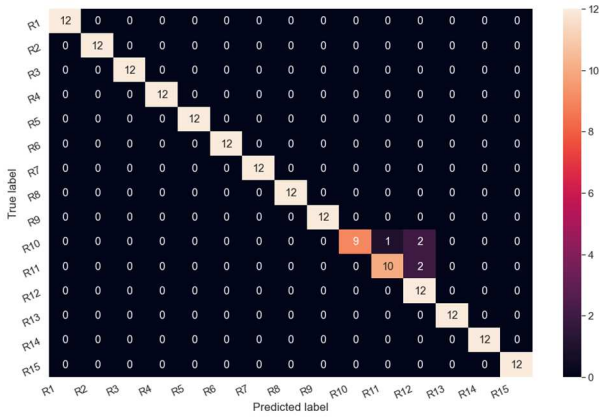

Fig.4: The RF confusion matrix is shown. The indices R_1-R_{15} signify the resistances connected at fifteen different locations (cf. Fig. 2). Testing data used=40%.

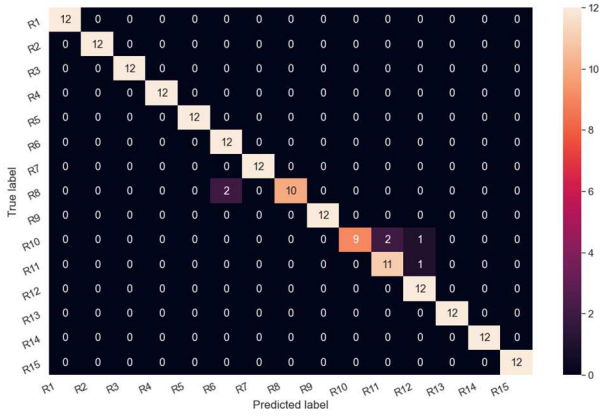

Fig.5: The XGBoost confusion matrix is shown.

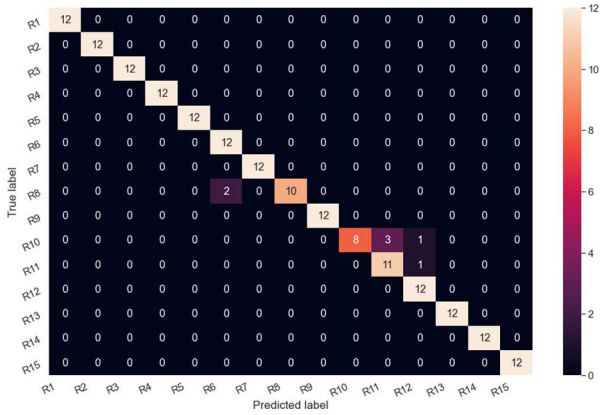

Fig.6: The LightGBM confusion matrix is shown.

III. Machine Learning Results

Based on the chosen algorithm models, a good overall classification accuracy on the testing data (even with low training data) is achieved as shown in the Table I. Of course, the accuracies of the algorithms depend on the training data percentage. The accuracy of the LightGBM algorithm remains reasonably high, even when only 30% data has been used for training. Fig. 4-6 show the confusion matrices calculated by the chosen algorithms, respectively. All samples, except for those from R_{10}, R_{11} are classified correctly as shown in Fig. 4. Fig. 5 shows how all but R_8, R_{10}, R_{11} are classified correctly. Similarly, as shown in Fig. 6, only R_8, R_{10}, R_{11} are wrongly classified. Interestingly, out of these three cases, R_{10} region (i.e. a resistance connected between V_R and V_{WL}) and R_{11} region (a resistance connected between V_R and V_{BL}) have fallen common.

The extracted features of importance are shown in Table II-IV, respectively. We find that, if one uses the XGBoost algorithm for our particular dataset, one would be able to maintain a good accuracy (cf. Table III) with the features obtained by DC analysis only. Intuitively, if one includes more extracted features performing conventional DC write mode of operations i.e. bias with V_{BL} (or V_{BLB})=V_{WL}=V_{dd}=1V, and V_{BLB} (or V_{BL})=V_{ss}=0V, the testing accuracy would be further improved.

Features	**Importance**
Rise time of V_L during transient analysis	0.105673
V_R=V_L during transient analysis	0.081829
Time when V_R=V_L during transient analysis	0.073452
Vary V_R during the read mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.066535
Vary V_L during the read mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.047744
Fall time of V_R during transient analysis	0.045796
Vary V_L during the retention mode of operation, find V_L when dV_R/dV_L=-1 for trip point voltage<V_L<V_{DD}	0.044916
Vary V_L during the retention mode of operation, find V_R when dV_R/dV_L=-1 for trip point voltage<V_L<V_{DD}	0.041186
Vary V_L during the retention mode of operation, find V_R when dV_R/dV_L=-1 for 0<V_L<trip point voltage	0.041043
Vary V_L during the read mode of operation, find V_R when dV_R/dV_L=-1 for 0<V_L<trip point voltage	0.040956

Table II: The important features obtained by RF algorithm.

IV. Conclusion

A way to systematically model the bridge-defects in a 6T-SRAM bit cell is explained which can provide a fast and effective way of studying the electrical behavior of the circuit with defects. Then, we have demonstrated an approach for predicting the locations of the defect in the circuit using a combination of HSPICE-generated defect database and machine learning. Our chosen set of algorithms, which are used as the predictive models, are trained with the electrical attributes calculated from the DC and transient response of the circuit. We have found that the proposed approach can achieve a fairly-high accuracy in predicting the defect location. Based on our approach, one can next try to predict the bridge-defect locations by incorporating multiple defects at a time. We believe that this ML-guided defect detection set up will further enhance the failure analysis success rate for more complicated circuits and systems.

Acknowledgment

This work is supported in part by A*STAR Accelerated Materials Development for Manufacturing Grant no: A1898b0043.

Features	Importance
Vary V_R during the retention mode of operation, find V_R when dV_L/dV_R=-1 for 0<V_R<trip point voltage	0.106309
Vary V_L during the retention mode of operation, find V_R when dV_R/dV_L=-1 for trip point voltage<V_L<V_{DD}	0.093608
Vary V_R during the retention mode of operation, find V_L when dV_L/dV_R=-1 for trip point voltage<V_R<V_{DD}	0.089980
Vary V_R during the read mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.082425
Vary V_R during the retention mode of operation, find V_L when dV_L/dV_R=-1 for 0<V_R<trip point voltage	0.068046
Vary V_R during the read mode of operation, find V_L when dV_L/dV_R=-1 for 0<V_R<trip point voltage	0.067159
Vary V_L during the read mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.065915
Vary V_R during the retention mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.061861
Time when V_R=V_L during transient analysis	0.054772
Vary V_L during the retention mode of operation, find V_R when dV_R/dV_L=-1 for 0<V_L<trip point voltage	0.051099
Vary V_L during the retention mode of operation, find V_L when dV_R/dV_L=-1 for 0<V_L<trip point voltage	0.046997
Rise time of V_L during transient analysis	0.041418

Table III: The important features obtained by XGBoost algorithm.

Features	Importance
Rise time of V_L during transient analysis	0.108148
V_R=V_L during transient analysis	0.075728
Vary V_R during the read mode of operation, find V_L when dV_L/dV_R=-1 for trip point voltage<V_R<V_{DD}	0.073770
Vary V_R during the retention mode of operation, find V_L when dV_L/dV_R=-1 for trip point voltage<V_R<V_{DD}	0.068021
Vary V_L during the retention mode of operation, find V_R when dV_R/dV_L=-1 for trip point voltage<V_L<V_{DD}	0.067776
Vary V_L during the read mode of operation, find V_R when dV_R/dV_L=-1 for trip point voltage<V_L<V_{DD}	0.067531
Fall time of V_R during transient analysis	0.062638
Vary V_R during the retention mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.051872
Vary V_L during the retention mode of operation, extract trip point (i.e. V_L=V_R) voltage	0.047101
Vary V_L during the retention mode of operation, find V_L when dV_R/dV_L=-1 for 0<V_L<trip point voltage	0.042696
Vary V_R during the retention mode of operation, find V_R when dV_L/dV_R=-1 for 0<V_R<trip point voltage	0.042696
Time when V_R=V_L during transient analysis	0.041228

Table IV: The important features obtained by LightGBM algorithm.

REFERENCES

[1] S. Mukherjee, S. Dutta, J. Ghosh, D. Saha, A. Laha, and S. Ganguly, "Temperature Dependent Variability Analysis of Threshold Voltage and On-Current for Optimum Switching Performance by Gallium Nitride-based Junctionless FinFET", IEEE Electron Devices Technology and Manufacturing Conference, pp. 118-120 (2019).

[2] J. Ghosh, V. Sverdlov, T. Windbacher, and S. Selberherr, "Spin Injection and Diffusion in Silicon Based Devices from a Space Charge Layer", Journal of Applied Physics 115, pp. 17C503 (2014).

[3] J.-S. Yoon, J. Jeong, S. Lee, and R.-H. Baek, "Systematic DC/AC Performance Benchmarking of Sub-7-nm Node FinFETs and Nanosheet FETs". IEEE Journal of the Electron Devices Society 6, 942-947 (2018).

[4] J. Ghosh and S. Ganguly, "Modeling and Simulation of AlGaN/InGaN/GaN Double Heterostructures using Distributed Surface Donor States", Japanese Journal of Applied Physics 57:8, 080305 (2018).

[5] W. Kim and L. Milor, "Built-in Self Test Methodology for Diagnosis of Backend Wearout Mechanisms in SRAM Cells", IEEE 32nd VLSI Test Symposium (2014).

[6] W. Kim, C.-C. Chen, D.-H. Kim, and L. Milor, "Built-In Self-Test Methodology With Statistical Analysis for Electrical Diagnosis of Wearout in a Static Random Access Memory Array", IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24:7, pp. 2521-2534 (2016).

[7] C.-W. Teo, "TCAD Modeling and Characterization of Defects in Advanced Nanoscale Semiconductor Devices", Master's Thesis, NUS Singapore (2019).

[8] C.-W. Teo, V. Narang, and A. V.-Y. Thean, "Electrical Characterization of FEOL Bridge Defects in Advanced Nanoscale Devices Using TCAD Simulations", IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits, pp. 1-4 (2018).

[9] SentaurusTM Process Explorer User Guide Version R-2020.09, Synopsys Inc. Sep 2020. Accessed on: May 20, 2020 [Online]. Available: https://spdocs.synopsys.com/dow_retrieve/R-2020.09/seg/sentaurus/pdf/spx_ug.pdf

[10] S. Y. Lim, J. Ghosh, A. Thean, "Innovative use of TCAD Process Simulation for Device Failure Analysis", IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (2021).

[11] Synopsys HSPICE circuit simulator https://www.synopsys.com/verification/ams-verification/hspice.html

[12] C.-W. Teo, K. L. Low, V. Narang, and A. V.-Y. Thean, "TCAD-Enabled Machine Learning Defect Prediction to Accelerate Advanced Semiconductor Device Failure Analysis", International Conference on Simulation of Semiconductor Processes and Devices, pp. 1-4 (2019).

[13] J. Pan, K. L. Low, J. Ghosh, S. Jayavelu, Md. M. Ferdaus, S. Y. Lim, E. Zamburg, Y. Li, B. Tang, X. Wang, J. F. Leong, S. Ramasamy, T. Buonassisi, C.-K. Tham, and A. V.-Y. Thean, "Transfer Learning-Based Artificial Intelligence-Integrated Physical Modeling to Enable Failure Analysis for 3 Nanometer and Smaller Silicon-Based CMOS Transistors", ACS Appl. Nano Mater. (2021).

[14] Y. S. Bankapalli; H. Y. Wong, "TCAD Augmented Machine Learning for Semiconductor Device Failure Troubleshooting and Reverse Engineering", International Conference on Simulation of Semiconductor Processes and Devices, pp. 1-4 (2019).

[15] K. Mehta, S. S. Raju, M. Xiao, B. Wang, Y. Zhang, H. Y. Wong, "Improvement of TCAD Augmented Machine Learning Using Autoencoder for Semiconductor Variation Identification and Inverse Design", IEEE Access 8, pp. 143519-143529 (2020).

[16] J. Lee, P. Asenov, M. Aldegunde, S. M. Amoroso, A. R. Brown, and V. Moroz, "A Worst-Case Analysis of Trap-Assisted Tunneling Leakage in DRAM Using a Machine Learning Approach", IEEE Electron Device Letters 42, pp. 156-159 (2021).

[17] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, accessed on 10th April 2021.

[18] https://xgboost.readthedocs.io/en/latest/, accessed on 10th April 2021.

[19] https://lightgbm.readthedocs.io/en/latest/, accessed on 10th April 2021.

[20] https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html, accessed on 10th April 2021.

[21] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, accessed on 10th April 2021.