Factors Prediction of Breakdown One-Time Programmable Gate-All-Around Junctionless Transistor Fabrication with Reinforcement Learning

Pei-Hung Chung
932002997
ECEN, Texas A&M University
chung95191@tamu.edu

Chun-Sheng Wu
832001192
CEEN, Texas A&M University
jinsonwu@tamu.edu

Abstract

The development of one-time programmable (OTP) memory brings a prospective future to the industries requiring rapid read accesses. Combining OTP memory with Gate-All-Around (GAA) technology can further bolster the circuit performance by achieving source/drain breakdown with growing active current. Employing junctionless (JLT) configuration on breakdown OTP-GAA memory solves electronics transmission issues with decreasing feature size. We can easily fetch the states of storage units with much less energy consumption through monitoring the drain/source current. However, the fabrication operations of breakdown are typically manual. In this situation, how to exploit machine learning techniques when executing breakdown is much of importance. Through the advancement of deep reinforcement learning (RL), uncertain fabrication factors are convincingly predictable. Provided that many properties of nano-scale devices are difficult to determine, rewarding rule and deep RL-based cut-off manager are beneficial for predicting these unclassified targets. In this project, we will perform reinforcement learning to predict key factors of the implementation of breakdown at our best.

1 Introduction

Over the past five years, semiconductor processing techniques have improved significantly. Due to the need of nano-scale advanced integrated circuits, the processing method employing on semiconductors with multiple gate channels is urgently required. Gate All-Around (GAA) is what applies to make current technology break through from FinFET (Fin Field Effect Transistor). GAA transistor enables multiple states accessing on a single body with multiple channels. Passing electronics through more channels can boost the probability of electronics transmission. Through this characteristics, the growing active current is able to resolve the predicament of severe current leakage while sustaining the same current scale. Meanwhile, multiple gate channels enhance the effective transistor density in a unit area.

With the emerging importance of data protection, one-time programmable (OTP) memory gradually stands out nowadays[1]. OTP memory has the potential of avoiding deliberate overwriting in numerous instructions, which is important when manufacturing read-only memory (ROM). To eliminate the static power consumption of accessing and improve the accessibility of ROM arrays, large I_d is especially needed. GAA is useful in increasing huge but unstable drain current. Accordingly, implementing breakdown on the areas between source/drain and gate becomes a solution[2]. This method is to exert large voltage stress on the thinnest internal connection between drain/source and gate[3]. The higher voltage stress imposed on the connection, the thinner the connection will be. The tendency of active current becomes linear after applying a large enough voltage, namely breakdown point. The electronic repulsive force between active regions and gate decreases swiftly after

Preprint. Under review.

breakdown. Active current accordingly has a substantial increase, which can be utilized to improve read access. Nevertheless, factors when executing breakdown on GAA transistors are difficult to determine. Having precise conditions of fabrication helps cut down the considerable experiment expenses. Hence, it is crucial to obtain the trade-off vector between fabrication factors to have the desired performance of OTP-GAA transistors.

However, under our best understanding, the methods to find the optimal value of breakdown points at present are still hand-crafted and experimental. Recently, people tend to settle a large range of sweeping voltage on active regions and pause the instrument manually after reaching breakdown points. In this experiment, we set up an interval of 0-20 volts and conducted the measurement. The timing of halting the instrument has an important contribution of deciding breakdown points. Nonetheless, controlling the apparatus by hand is possible to have experimental deviation. Redundant fabricating operations lead to unignorable energy consumption as the circuits are scaled up. On the other hand, precise breakdown points can assists us to fabricate OTP transistors with less time. In that case, we proposed this project to utilize machine learning on determining demanding factors of fabrication. To be more concise, we aimed to employ machine learning methods on the prediction of breakdown points and its possible curves. Furthermore, our plan was to adopt a deep neural network based model to find the best tendency. To have better performance, further reinforcement learning techniques are exploited to strengthen the model, assisting semiconductor companies reduce the unnecessary experimental effort and earn more profit.

2 Related Work

OTP memory is typically produced using fuse or antifuse. Antifuse can be programmed by operating breakdown on the connection of oxide and source/drain area. It creates conductive path once the supplied voltage exceeds the limit[4][5]. Multistate OTP memory has appeared in the work of recent field effect transistor fabrication[6]. In this project, we will perform original (00), source breakdown (01), drain breakdown (10), and totally breakdown (11) on the device[7]. Under different conduction path, the active current increments related to the size of electronics transmission[8]. **Fig. 1** exhibits the location where breakdown takes place.

To reinforce the current density in a single transistor, GAA nanowire is universally adopted to produce novel architecture[9][10]. While devices decreasing sizes, JLT configuration can easily control the dopant profile in nano-scale[11]. Being capable of improving interface state density and short-channel effect (SCE) immunity lets JLT a great candidate for fabricating multistate OTP memory[12][13]. SCE also happens with decreasing channel length. GAA structure has high active current, multiple gate channels, and reduced channel doping concentrate to better resolve SCE than planar structure[14]. With above perspectives, the device in the experiment was fabricated through poly-silicon (poly-si) GAA JLT configuration.

Regardless of the success of GAA techniques in the chip fabrication field, the timing of the cut-off drain current still needs to be decided in more smart and automatic methods to save efforts. Here, we aim to introduce RL-based cut-off manager, offering a opportunity for the semiconductor industry to achieve optimal cut-off current without human inspection. Hence, that is why the manufacturing process with deep learning methods come in handy.

In previous approaches, the experts choose the best cut-off current based on the some pre-defined rules without randomness. In this paper, a more flexible cut-off manager is considered, which is quite different from the previous work. Here the neural network-based manager is learned by DQN[15] from scratch without any labeled data for pre-training. We used the advanced DQN algorithms[16] to improve the performance of the manager.

We will elucidate the reinforcement learning algorithm we used in section 3. In section 3.1, we illustrate the Markov Decision Process, as how we model the cuf-ff current problems. In section 3.2, we mention how the Deep Q Network algorithm works. In section 3.3, we introduce two other advanced algorithms which are similar as the original Deep Q Network. In section 3.4, we show the ideas of how we design the Deep RL-based cut-off manager by Markov Decision Process.

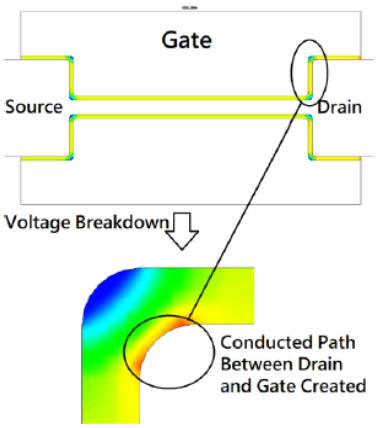

Fig. 1 displays where we perform the voltage during breakdown. The connection will obtain large short current ($10^{-}3A$) after the conducted path is formed[8].

3 Approach

There are several attributes for OTP-GAA transistors. The DNN-model and RL utility opens a breakthrough point that makes the breakdown points prediction based on the electric properties available. To satisfy the conditions of executing Machine Learning strategies, we should first have the data preprocessing. The data preprocessing was handled by dropping redundant information and the rearrangement according to fabrication factors. Secondly, we extract the feature vectors from the attributes by using the pretrained model. These feature vectors undergo categorization and normalization to fit in the model. We additionally process the data augmentation to deal with the lack of featurization. Third, we treat the processed feature vector as input and the output would be the breakdown tendency curve and breakdown points depending on the fabrication factors. The machine can predict an approximate breakdown point with a small voltage swing and cut-off the monitoring itself. Through the rewarding rule of RL, positive and negative feedback help the model converge to the optimal weights. Next, compares the predictions with the fabrication factors (input) to seek a best set of factors. By doing so, we can have the most favorable fabrication environment to produce the desired GAA FET without countless attempts. Meanwhile, manipulates the experiment by machine itself can reduce avoidable energy consumption and deviation causing by human operation. In the following, we will demonstrate the RL techniques we adopted.

3.1 Markov Decision Process

In order to solve the optimal cut-off ratio problem, we model the decision process as a Markov Decision Process(MDP).[17, 18, 19] A MDP is defined as a tuple $\{S, A, T, R, \gamma\}$, where S is the set of states, A the set of actions, R the reward function, T the transition probabilities and γ the discount factor. A mapping from a state $s \in S$ to an action $a \in A$, or action selection at each state, is a policy π. Given a policy π, the value of the Q-function ($Q^{\pi} : S \times A \rightarrow R$) is defined as the expected discount sum of all rewards that can be received by an agent over an infinite state transition path starting from state s taking action a:

$$Q^{\pi}(s, a) = E[\sum_{k=0}^{\infty} \gamma^{k} r_k | s_0 = s, a_0 = a], \tag{1}$$

where r_k is the reward received from the action a_k taken at state s_k, where k is the sequence index for states and actions. The optimal policy maximizes the value of each state-action pair: $\pi^*(s) = \arg\max_{a \in A} Q^*(s, a)$, where

$$Q^*(s, a) = E_{s'|s,a}[R(s, a, s') + \gamma \max_{b \in A} Q^*(s', b)] \tag{2}$$

and $R(s, a, s')$ is the reward for taking action a at state s and transit to state s'. Equation (2) is known as the Bellman optimality equation. Therefore, finding an optimal policy is equivalent to find the optimal Q-function which can be solved iteratively known as *Value Iteration*. The Q-function can be just a simple function in Section 3.1, or estimated by a neural network in deep reinforcement learning in Section 3.2.

3.2 Deep RL

We applied Deep-Q-Network algorithm here to optimize the Q-function. The DQN is a feed forward neural network [20, 21]with parameters θ to estimate the state-action value function $Q(s, a; \theta)$[1]. The input of the DQN is the feature of the GAA chips at the specific time step, including the instantly exact drain current, tendency of previous drain current, instantly exact voltage, tendency of previous, and other attributes of the corresponding experiment settings. On the other hand, the output of the DQN is the state-action value $Q(s, a; \theta)$ for each action a. Here, the output dimensions of the DQN are binary, indicating the probability for machine to execute the cut-off ot to stay put. The DQN is trained by iteratively updating the parameters θ. With parameters obtained at the i-th iteration, denoted as θ^i, θ^i can be learned by minimizing the following loss function $L_i(\theta^i)$ in (3) by gradient descent.

$$L_i(\theta^i) = \underset{s,a,r,s' \sim \mathcal{U}(\mathcal{T})}{E} [(\hat{y}_i - Q(s, a; \theta^i))^2]. \tag{3}$$

$\mathcal{T} = \{e_1, e_2, ..., e_t, ...e_T\}$ used for training includes experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ (taking action a_t at state s_t obtaining reward r_t and reaching state s_{t+1} at the next time step). $\mathcal{T}$ is a data set collected from many retrieval episodes. The expression $s, a, r, s' \sim \mathcal{U}(\mathcal{T})$ in (3) means the network is trained by sampling mini-batches of experiences from $\mathcal{T}$ uniformly at random. This method is referred to as experience replay, which is a key ingredient behind the success of DQN. In this way, the efficiency in using the training data can be improved through re-use of the experience samples in multiple updates, and the correlation among the samples used in the update can be reduced through the uniform sampling from the replay buffer.

$\hat{y}_i$ in (3) is defined as below:

$$\hat{y}_i = r + \gamma \cdot \max_{a' \in \mathcal{A}} Q(s', a'; \theta^-) \tag{4}$$

where θ^- represents the parameters of a fixed and separate target network. Equation (4) serves as the *sampled* version of (2) handling the problem of unknown transition probabilities. Here θ^- is taken as the parameters obtained several iterations before. Freezing the parameters of the target network $Q(s', a'; \theta^-)$ for a fixed number of iterations while updating the online network $Q(s, a; \theta^i)$ is another key innovation for the success of DQN, which improved the stability of the algorithm.

3.3 Extensions of Deep RL

Besides standard DQN algorithm, the extension versions are used. The standard DQN algorithm is noted to overestimate the Q-values[22], so Double DQN [23]is also used in the following experiment, which decouples the selection from the evaluation. Dueling DQN is further applied here.

Compared to typical DQN, dueling DQN [24] uses an altered network structure, splitting it into two streams: one learns to provide an estimate of the state value $V(s)$ for every state, and the other calculates the potential advantages of each action at a given state.

3.4 Deep RL-based Cut-Off Manager

Given the electric properties and the current-voltage pair at present as the input of Cut-off Manager, we use a Markov Decision Process(MDP) to model the prediction problem. As the breakdown point of the GAA transistors is not absolute,without ground truth existed, we defined the action as execution cut-off or not. Furthermore, we desire to make machine learn the optimal cut-off point in a more precise way. Not only an output vector based on stochastic prediction, instead, we aim to make DQN agent learned what should have done in every single time step. The state feature, action space, and reward system are defined in the following sections.

[1]Because the state-action value function $Q(s, a)$ here depends on the network parameters θ, the function is written as $Q(s, a; \theta)$.

3.4.1 State feature

We divide the feature into two different state feature: static state and dynamic state. Static state indicates the characteristics of the GAA, which will be fixed in the training process, including electric properties, the exact order of the manufacturing process, or the equipment settings of the experiment. On the other hand, dynamic state refers the current-voltage pair at the present, the instant condition of the chip, and the historical state of the aforementioned feature.

We experimented with these two types of feature as our state of the MDP setting, we designed the neural network model with one encoder and decoder, and extracted the latent representation as k-dimensional states. The value k is still to be decided here.

3.4.2 Action Space

The goal of the proposed cut-off manager is to offer an optimal breakdown point when the drain current increases. For each time step, or say each data point of drain current,the network would recommend to cut-off should be set on or off, so the action space of the QDN is a binary vector.

3.4.3 Reward System

The design of the reward system in a reinforcement learning based model is always crucial and tricky. Here, we simply define the reward value by intuition. Since the cut-off current handcrafted by expert is not absolute, we regard the cut-off is successful if the cut-off point appears close enough to the expert annotated point. We define close enough as the cut-off point is located in the 10 data point near the expert annotated truth. The nearest the cut-off point is, the reward it will received after the cut-off. If it succeeds, the agent will receive at most 10 positive reward; and if it fails, the agent will receive -10 negative reward as the penalty. That is to say, if the cut-off point appears near the expert annotated point from 3 data points, the agent will receive a 7 positive reward. When the process terminated since the drain current could not be able to increase, a negative -100 reward would be given to the end.

4 Dataset

4.1 Device Fabrication

Our data was measured on antifuse OTP-GAA poly-silicon junctionless nMOS[25]. Its fabrication process first applied low-pressure chemical vapor deposition (LPCVD) on the deposition of etching stopping layers with Si_3N_4. A silicon-nitride layer was then deposited by selective deposition with LPCVD as the diffusion stopping layer. Next, I-line based projection lithography was employed for dummy region masks, providing the prerequisites of source/drain regions. Subsequently, a tetraethoxysilane (TEOS) oxidized layer was deposited as source/drain regions to be wet etched later for suspending gate regions. We adopted reaction ion etching (RIE) with TEOS sidewall protection to implement selective anisotropic etching on the Si_3N_4 at bottom. Operated LPCVD to deposit a phosphorous-doped amorphous-silicon (single-silicon) layer stacking on the TEOS layer. Afterwards, the single-silicon layer was transformed to poly-silicon layer by second solid phase crystallization (SPC). Utilized phosphoric acid to remove the dummy mask. At last, deposited a silicon-nitride on the top of the TEOS layer performing as the gate nanowire channels[26]. **Fig. 2** illustrates the steps of fabrication. Because we fabricated the device in a 6-in silicon wafer, the dies predictably had some different characteristics causing by the location with the same process.

4.2 Data Acquisition

In this experiment, there were different dies, fabricating locations, and constant gate voltage (V_g) as variables. During the measurement, we first observed its original electric properties by fetching active current to drain voltage (I_d/I_s-V_d) and active current to gate voltage (I_d/I_s-V_g) curves. We conducted two kinds of breakdown strategies, drain breakdown first and source breakdown first respectively. By imposing large scales of voltage on source/drain region with floating another active region, the changing drain current displayed the status of breakdown. **Fig. 3** shows the status of active current when exerting growing drain voltage on the connection. It reached the breakdown point while active dramatically increased and sustained for a 1-volt interval. After active current became stable, paused

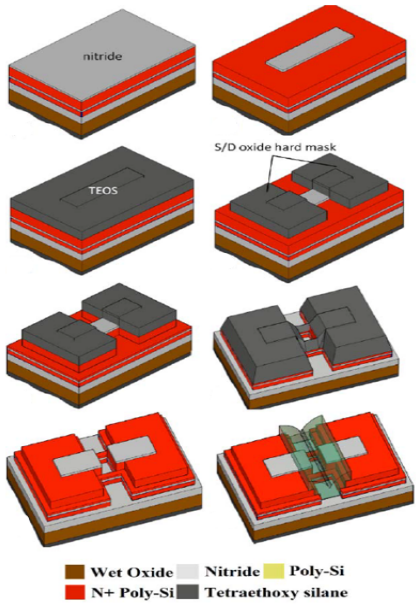

Fig. 2 displays the steps of fabricating this antifuse OTP-GAA poly-silicon junctionless nMOS in 1-μm scale[8].

the imposing voltage and observed the breakdown points and curve tendency. Then, remeasured the active current to drain voltage and gate voltage curves to inspect the difference between original status. Meanwhile, in order to verify its functionality of repeat accesses, the reversed active current to drain voltage and gate voltage curves were documented as well. Lastly, performed breakdown on the opposite side (source/drain) and observed the electric properties again and repeatedly measured the above data on all the testing dies.

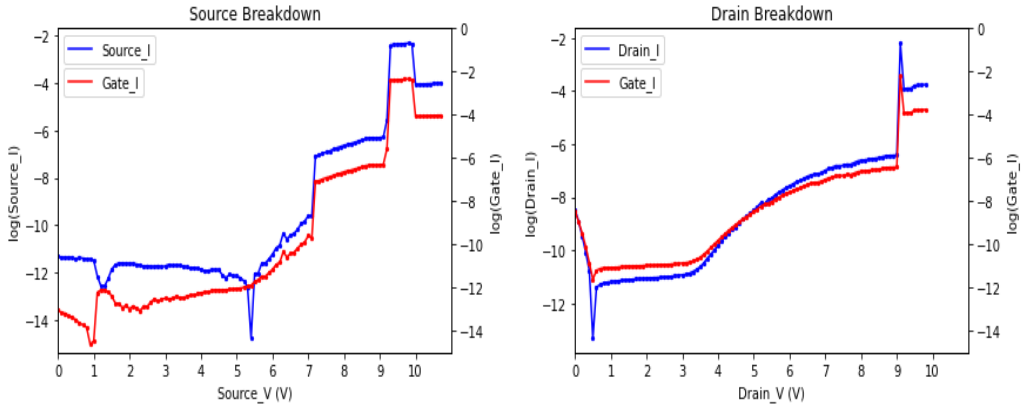

Fig. 3 displays the values of I_d/I_s while we keep increasing V_d. The left graph is drain breakdown and the right one is source breakdown. From the graph, we can clearly observe that I_d has a significant increase after a specific point, which is so called breakdown point.

4.3 Training Dataset Description

From the above measurement, we collected a total of 66748 data points in 11 features as training dataset. The features are respectively V_d, I_d/I_s, I_g, V_g, Breakdown order, die number, location, width, length, breakdown point on drain, and breakdown point on source. Each feature in each graph is recognized as a contribution in RL training. These data compose 96 different graphs based on 12 measuring steps. Graphs contained 2 superimposed curves inside, totally 192 independent curves used for tendency prediction.

Table 1: Accuracy and Return for cut-off manager by different approaches

$Methods$	DQN	$DoubleDQN$	$DuelingDQN$	Random
Returns	3.42	2.50	3.56	-7.34
Accuracy	0.0563	0.0243	0.0312	0.0141
Accuracy-6	0.1548	0.1007	0.1787	0.0833
Accuracy-10	0.1924	0.2085	0.2159	0.1408
Accuracy-30	0.3064	0.2937	0.2980	0.4225

4.4 Data Preprocessing

In case that the raw data was abstruse to directly apply by machine, we firstly abandoned the unnecessary features, e.g. instrument labels and time lapses. Secondly, recategorized the data by its fabrication factors, electric properties (I_d/I_s-V_d and I_d/I_s-V_g), breakdown tendency, and breakdown points. Additionally, to have sufficient training data, we reorganized the previous data according to the steps of implementing breakdown. Next, superimposed noise on the data to benefit the training performance. The superimposed data was then formatted to fit in with the input of our model. Ultimately, separated the whole data recursively by the input format of our model. This technique would augment the performance of RL rewarding rule and found the optimal solution more rapidly.

5 Experiment

5.1 Experimental setting

Our cut-off manager is based on Deep-Q-Network and its variants. The input dimension K of the decision makers were set to 11. The output dimensions for the DQNs in the cut-off managers were all binary.

All the DQNs have the same hyper-parameters. We used network with two hidden layers of 1024 nodes. We used relu as the activation function and set the batch size to 256. Initial learning rate is set as 8e-4. Accuracy was selected as our cut-off performance evaluation metric. Ten-fold cross validation was performed in all experiments, that is, for each trial, 8 out of 10 query folds were used for training, 1 for parameter tuning (validation set), and the remaining fold for testing.

5.2 Result and discussion

As you can see in the first row of table 1, we apply DQN, double DQN, and dueling DQN as our cut-off manager. The second row of table 1 shows the value the Returns, where Returns indicates the sum of the rewards in average that cut-off manager received in each iteration. In addition, the rest of rows are Accuracy and Accuracy-N. Accuracy indicates the percentage of cut-off manager predict the exact current step same as the expert. On the other hand, Accuracy-N indicates the percentage of cut-off manager predict the relative current step in the N current step interval.

The Random in the last column of Table 1 indicates the cut-off manager choose the action randomly, and we regard the Random manager as our baseline. For different evaluation metrics, we can see the RL-based cut-off manager outperforms the baseline in both Returns, Accuracy, Accuracy-6. For Accuracy-30 in Table 1, the result show that all the RL-based manager couldn't reach the Random manager level. We guess the reason why the RL-based mamager couldn't works as good as the Random one is the manager learn to avoid the large penalty when the session terminate.

For the different algorithm in RL, both of them act roughly the same in all settings. Notably, Double DQN performs a little worse in Returns, Accuracy, Accuracy-6 and Accuracy-10 than the other two cut-off manager and works roughly equally in Accuracy-10, comparing with the DQN and dueling DQN.

6 Conclusions

In this paper, we proposed the automatic decision-making cut-off manager based on reinforcement learning methods. Under our best knowledge, this is the first attempt to predict the optimal cut-off current of GAA among the semiconductor research field. It's no secret that the data acquired from the semiconductor manufacturing laboratory or industry is really expensive, so even we couldn't have obtained promising enough performance to apply on the real manufacturing process because of the precious little data, we believe the idea will still work and even work much better if we could obtain more attributes and characteristics in the manufacturing process.

7 Reference

[1] Jin-Woo Han, Dong-Il Moon, and M. Meyyappan. "One Time Programmable Antifuse Memory Based on Bulk Junctionless Transistor". In: *IEEE Electron Device Letters* 39.8 (2018), pp. 1156–1158. DOI: 10.1109/LED.2018.2848975.

[2] D.K.Y. Liu et al. "Scaled dielectric antifuse structure for field-programmable gate array applications". In: *IEEE Electron Device Letters* 12.4 (1991), pp. 151–153. DOI: 10.1109/55.75747.

[3] Shen-Yang Lee et al. "Experimental Demonstration of Stacked Gate- All-Around Poly-Si Nanowires Negative Capacitance FETs With Internal Gate Featuring Seed Layer and Free of Post-Metal Annealing Process". In: *IEEE Electron Device Letters* 40.11 (2019), pp. 1708–1711. DOI: 10.1109/LED.2019.2940696.

[4] Kensuke Matsufuji et al. "A 65nm Pure CMOS one-time programmable memory using a two-port antifuse cell implemented in a matrix structure". In: *2007 IEEE Asian Solid-State Circuits Conference*. 2007, pp. 212–215. DOI: 10.1109/ASSCC.2007.4425768.

[5] Zicheng Liu, Ruifeng Zheng, and Jianwei Sun. "A gate-oxide-breakdown antifuse OTP ROM array based on TSMC 90nm process". In: *2015 International Symposium on Next-Generation Electronics (ISNE)*. 2015, pp. 1–3. DOI: 10.1109/ISNE.2015.7132015.

[6] Po-Jung Sung et al. "High-Performance Uniaxial Tensile Strained n-Channel JL SOI FETs and Triangular JL Bulk FinFETs for Nanoscaled Applications". In: *IEEE Transactions on Electron Devices* 64.5 (2017), pp. 2054–2060. DOI: 10.1109/TED.2017.2679766.

[7] Seung-Youl Kim et al. "Design of CMOS dual antifuse OTP memory based on gate oxide". In: *2014 International SoC Design Conference (ISOCC)*. 2014, pp. 284–285. DOI: 10.1109/ISOCC.2014.7087650.

[8] Chen-Feng Chang et al. "Investigation of Two Bits With Multistate Antifuse on nMOS Poly-Silicon Junctionless GAA OTP". In: *IEEE Transactions on Electron Devices* (2021), pp. 1–6. DOI: 10.1109/TED.2021.3122885.

[9] Michele De Marchi et al. "Configurable Logic Gates Using Polarity-Controlled Silicon Nanowire Gate-All-Around FETs". In: *IEEE Electron Device Letters* 35.8 (2014), pp. 880–882. DOI: 10.1109/LED.2014.2329919.

[10] Yao-An Chung et al. "Novel hybrid 3D NAND flash memory containing vertical-gate and gate-all-around structures". In: *2016 27th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. 2016, pp. 371–374. DOI: 10.1109/ASMC.2016.7491079.

[11] Ya-Chi Cheng et al. "Characteristics of a Novel Poly-Si P-Channel Junctionless Thin-Film Transistor With Hybrid P/N-Substrate". In: *IEEE Electron Device Letters* 36.2 (2015), pp. 159–161. DOI: 10.1109/LED.2014.2379673.

[12] Lun-Chun Chen et al. "Junctionless Poly-Si Nanowire FET With Gated Raised S/D". In: *IEEE Journal of the Electron Devices Society* 4.2 (2016), pp. 50–54. DOI: 10.1109/JEDS.2016.2514478.

[13] Chi-Woo Lee et al. "Junctionless multigate field-effect transistor". In: *Applied Physics Letters* 94.5 (2009), p. 053511. DOI: 10.1063/1.3079411. eprint: https://doi.org/10.1063/1.3079411. URL: https://doi.org/10.1063/1.3079411.

[14] Nima Dehdashti Akhavan et al. "Improvement of carrier ballisticity in junctionless nanowire transistors". English. In: *Applied Physics Letters* 98.10 (Mar. 2011). ISSN: 0003-6951. DOI: 10.1063/1.3559625.

[15] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[16] Matteo Hessel et al. “Rainbow: Combining Improvements in Deep Reinforcement Learning”. In: (2017). cite arxiv:1710.02298Comment: Under review as a conference paper at AAAI 2018. URL: http://arxiv.org/abs/1710.02298.

[17] Richard Bellman. *Combinatorial processes and dynamic programming*. Tech. rep. RAND CORP SANTA MONICA CA, 1958.

[18] Richard Bellman. “A Markovian decision process”. In: *Journal of mathematics and mechanics* 6.5 (1957), pp. 679–684.

[19] Stuart Dreyfus. “Richard Bellman on the birth of dynamic programming”. In: *Operations Research* 50.1 (2002), pp. 48–51.

[20] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[21] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[22] Hado Van Hasselt, Arthur Guez, and David Silver. “Deep reinforcement learning with double q-learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[23] Hado Hasselt. “Double Q-learning”. In: *Advances in neural information processing systems* 23 (2010), pp. 2613–2621.

[24] Ziyu Wang et al. “Dueling network architectures for deep reinforcement learning”. In: *International conference on machine learning*. PMLR. 2016, pp. 1995–2003.

[25] Yu-Zheng Chen et al. “Multilevel Anti-Fuse Cells by Progressive Rupturing of the High-κ Gate Dielectric in FinFET Technologies”. In: *IEEE Electron Device Letters* 37.9 (2016), pp. 1120–1122. DOI: 10.1109/LED.2016.2591581.

[26] Chun-Jung Su et al. “Gate-All-Around Junctionless Transistors With Heavily Doped Polysilicon Nanowire Channels”. In: *IEEE Electron Device Letters* 32.4 (2011), pp. 521–523. DOI: 10.1109/LED.2011.2107498.