information

Article

CAPTIVE: Constrained Adversarial Perturbations to Thwart IC Reverse Engineering

Amir Hosein Afandizadeh Zargari [1,†], Marzieh AshrafiAmiri [1,†], Minjun Seo [1], Sai Manoj Pudukotai Dinakarrao [2,*], Mohammed E. Fouda [1] and Fadi Kurdahi [1]

1 Department of Electrical and Computer Engineering, University of California, Irvine, CA 92697, USA; amir.zargari@uci.edu (A.H.A.Z.); mashrafi@uci.edu (M.A.); minjun.seo@uci.edu (M.S.); foudam@uci.edu (M.E.F.); kurdahi@uci.edu (F.K.)
2 Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030, USA
* Correspondence: spudukot@gmu.edu; Tel.: +1-703-993-1375
† These authors contributed equally to this work.

Abstract: Reverse engineering (RE) in Integrated Circuits (IC) is a process in which one will attempt to extract the internals of an IC, extract the circuit structure, and determine the gate-level information of an IC. In general, the RE process can be done for validation as well as Intellectual Property (IP) stealing intentions. In addition, RE also facilitates different illicit activities such as the insertion of hardware Trojan, pirating, or counterfeiting a design, or developing an attack. In this work, we propose an approach to introduce cognitive perturbations, with the aid of adversarial machine learning, to the IC layout that could prevent the RE process from succeeding. We first construct a layer-by-layer image dataset of 45 nm predictive technology. With this dataset, we propose a conventional neural network model called *RecoG-Net* to recognize the logic gates, which is the first step in RE. RecoG-Net is successful in recognizing the gates with more than 99.7% accuracy. Our thwarting approach utilizes the concept of adversarial attack generation algorithms to generate perturbation. Unlike traditional adversarial attacks in machine learning, the perturbation generation needs to be highly constrained to meet the fab rules such as Design Rule Checking (DRC) Layout vs. Schematic (LVS) checks. Hence, we propose CAPTIVE as a constrained perturbation generation satisfying the DRC. The experiments show that the accuracy of reverse engineering using machine learning techniques can decrease from 100% to approximately 30% based on the adversary generator.

Keywords: reverse engineering; Integrated Circuits; adversarial attacks; machine learning

check for updates

Citation: Zargari, A.H.A.; AshrafiAmiri, M.; Seo, M.; Pudukotai Dinakarrao, S.M.; Fouda, M.E.; Kurdahi, F. CAPTIVE: Constrained Adversarial Perturbations to Thwart IC Reverse Engineering. *Information* **2023**, *14*, 656. https://doi.org/10.3390/info14120656

Academic Editor: Paolo Maistri

Received: 1 November 2023
Revised: 29 November 2023
Accepted: 5 December 2023
Published: 11 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

To meet the large operational, maintenance, and development costs, the semiconductor industries are inclining towards a fabless business model, i.e., outsourcing the fabrication to offshore foundries. Such outsourcing has also led to other benefits of outsourcing the Integrated Circuit (IC) fabrication and adopting a global supply chain for reduced capital and maintenance costs, minimized design-flow efforts, and time-to-market [1,2]. Despite the achieved benefits, such outsourcing and adoption led to a complex verification and fabrication cycle, and supply-chain increased possible hardware threat space, arising in different forms, namely IC piracy, overproduction, hardware Trojan (HT) insertion, and reverse engineering [3–5].

Reverse Engineering (RE) is a process in which one attempts to extract the internals of an IC, extract the circuit structure, and determine the gate-level information of an IC [6–12]. The RE can lead to adversarial consequences, including IP theft and IP stealing, eventually leading to financial losses [13]. Among multiple reverse engineering techniques, imaging-based reverse engineering [14,15] is one of the prominent threats that cannot be mitigated, as fabricated devices are available in the market for consumer and commercial systems. The existing imaging-based reverse engineering methods can be divided into destructive and non-destructive approaches [6]. In the destructive method, first Scanning Electron Microscopy (SEM) images of different layers of the layout are captured. The obtained IC layout or SEM images are fed to reverse engineering tools such as DeGate [14] for reverse engineering and annotation. In contrast, the foundries can also adapt other recently introduced non-destructive imaging techniques such as Ptychographic X-ray laminography, ensuring the reverse-engineered IC functions and not destructed [16].

The imaging-based reverse engineering techniques have two steps in common. First of all, the gates inside each design should be annotated, then by finding the connectivity between the gates, the whole reverse engineering process is completed, and the layout design can be revealed. Given the success of machine learning and statistical methods in a wide range of applications [17–21] with computer vision and hardware security being no exceptions, we develop a machine learning model; RecoG-Net is capable of recognizing the type of gate by using the image of one layer of the layout.

Further, as our principal contribution, we propose CAPTIVE, Constrained Adversarial Perturbations to Thwart IC Reverse Engineering. CAPTIVE is capable of making gate recognition, the first step of reverse engineering, impossible. In the CAPTIVE method, we introduce Design-Rule-Checking (DRC)-compliant adversarial perturbations which are specially crafted noises and are sometimes invisible to human eyes. The main challenge was to convert the existing adversarial noises [22–24] to DRC-compliant objects that can be added to the layout and also be fabricated. An approach is introduced towards reaching these specific DRC-compliant perturbations. To validate the effectiveness of the CAPTIVE method, DRC-compliant objects are added to the SEM images of layer(s), and the experiments indicate that the RecoG-Net's performance in gate recognition will drastically drop up to 70% (from near 100% to 30%).

The reported results present that even with finding the perfect connectivity if an attacker can not recognize gates, the process of reverse engineering will fail.

The contributions of this work can be outlined in two main points as follows:

- We propose RecoG-Net, a convolutional neural network model, to fully recognize the gates from single/multiple layer(s) of SEM or the layout image(s) with approximately 100% accuracy.
- We propose CAPTIVE as a method to add DRC-complaint perturbation to layout images to thwart IC-RE. We perform different experiments to validate the efficiency of CAPTIVE.

The proposed CAPTIVE methodology can enable the design of secure and invasive/optical reverse engineering resilient ICs by the inclusion of adversarial perturbations. Further, to standardize the process of the inclusion of the adversarial perturbations in the IC design, the perturbation information can be inserted in the standard cell libraries, similar to how the LP and HP cells are defined in the cell libraries. Such inclusion can enable less complex and non-intrusive inclusion of the adversarial perturbations in the IC layouts without the need for disrupting the EDA design flow and yet enable resilient IC designs compatible with existing fabrication processes.

The rest of this paper is organized as follows: Section 2 describes the reverse engineering process and our data generation mechanism. Moreover, the RecoG-Net model has been proposed which is responsible for performing the gate recognition. Section 3 focuses on the CAPTIVE method to make the first step of reverse engineering, gate recognition, less possible. The evaluation of the proposed CAPTIVE method and some relevant background regarding the existing reverse engineering methods are presented in Section 4. Section 5 contains the conclusion and the future work.

2. IC Reverse Engineering Attack

Nowadays, the fabricated ICs are vulnerable to different malicious behaviors. One of the main possible threats is reverse engineering on the fabricated IC. To perform the process of reverse engineering on the chip and extract the internal structure inside it, first, gates, which are the small components inside the chip, should be determined. Then, by finding the connectivity between different gates, the logic and internal structure of the IC can be determined.

As our first contribution, we propose the RecoG-Net model which will help us in classifying gates and preventing the first step of IC reverse engineering. To achieve this, first, we need to create a specific dataset of different existing gates.

2.1. Attack Model

Integrated Circuit (IC) design refers to a process of assembling a collection of circuit elements like transistors, resistors, and capacitors to perform a specific function. These components are combined to form more complex functions such as logic gates, which are then connected to build more complex segments such as adders and multipliers. This process continues to build on itself, resulting in the availability of increasingly complex circuit building blocks.

In IC design, the circuit elements are implemented on a silicon substrate using a process called photolithography. The photolithography process creates various geometric shapes on the silicon substrate where the electrical properties of the region defined by that shape are altered. Basic circuit elements are created when these regions are combined and superimposed over each other.

Thus, IC design consists of two distinct processes. First, circuit elements are gathered together in one place to perform some specific pre-defined function. Next, the various geometric shapes that implement those circuit elements must be assembled and interconnected on the silicon substrate. The first process is typically called logic design, and the second process is called physical design.

Nowadays, several companies accomplish the logic design; but these companies remain fabless due to the high cost of physical design. Therefore, these companies will send the designed circuit to the fabrication company for physical design, which introduces additional challenges, especially reverse engineering, as aforementioned.

The threat model we consider in this work is as follows: The attacker has access to a fully functional IC and the goal is to extract the netlist and the internals of the design. In order to achieve this, the attacker performs a destructive de-layering of the design [3,6,25–27]. Based on the de-layered images of the IC, the attacker feeds it to an automated tool such as DeGate [28] or a similar surrogate reverse engineering tool to annotate the gates and extract the high-level information of the IC. A crucial step in the attack process is to identify the individual gates and determine the interconnectivity to exploit the IP further. This process is also outlined in Figure 1.

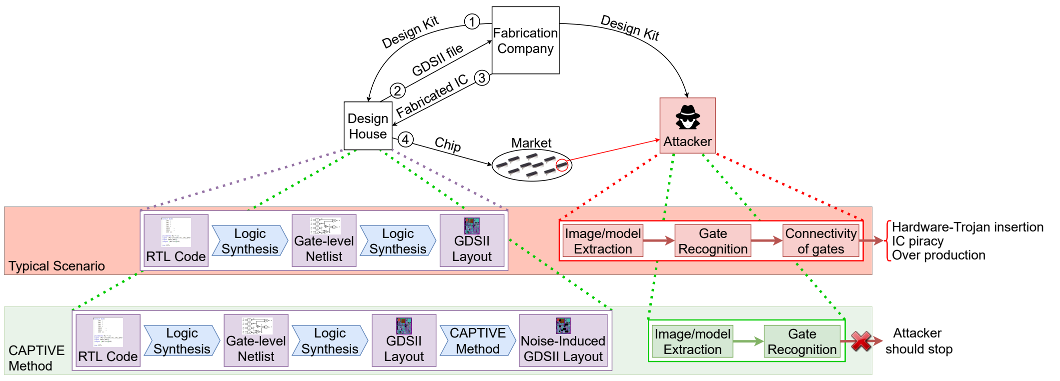

Figure 1. IC reverse engineering attack with and without CAPTIVE.

2.2. RecoG-Net: Surrogate Reverse Engineering Model

The primary next step towards reaching our goal is finding a method to determine the type of gate using the image(s) of one or multiple layers. Since each of these images contains a tremendous amount of information about the gate, using machine learning techniques seems to be reasonable. Machine learning techniques are capable of learning the patterns in the data, based on which, they can classify or predict a particular outcome for another set of input data during inference.

We propose a recognition network, RecoG-Net, a combination of two powerful machine learning techniques, Convolutional Neural Network (CNN) with fully connected layers. As per the literature, such networks are efficient for having an automatic feature selection, which will help reduce the dimension of our input. The dimensionality reduction procedure will both preserve features that are mostly related to the specific characteristics of a gate and eliminate the meaningless ones. This will help the network to learn valuable features thoroughly.

Our experimental dataset consists of 889 images with a size of 258 by 1049, which are converted into black-and-white images to obtain maximum contrast. The dataset is divided into two parts for the training and testing phases, with 700 and 189, respectively. The structure of the model that we have proposed is illustrated in Table 1. First of all, using a 2D-convolutional layer and 32 filters of size 3 × 3, the 258 × 1049 features of the data point are converted to a matrix of 1256 × 1047 × 32. This initial layer will learn the basic features of the data. The second 2D convolutional layer, which will help learn more complex features, is being used with a similar structure. The pooling layer will slide a filter size of 3 × 3 across the 254 × 1045 × 64 features and replace it with the maximum value. Therefore, it will result in discarding 45% of the features in the matrix, shaped 84 × 348 × 64. To lower the possibility of overfitting, a dropout layer is utilized. The dropout rate is equal to 0.5 in our case, which means that 50% of the features will be discarded. Another set of convolutional layers and pooling is used for understanding even more complex features. Again, to reduce the likelihood of overfitting, a dropout layer of 0.5 is applied. A flattened layer is used to convert the 3-dimensional matrix with the size of 26 × 114 × 64 to a 1-dimensional matrix of size 189,696. Then, a fully connected layer of size 250 is followed by the convolutional network. In the end, a dense linear layer with a size equal to 11 will produce the predicted output. More detailed explanations of each layer can be found in Table 1. To choose the filters, first, we assign a filter size. Then, CNN randomly initializes the filters and trains until it reaches the best filter. Then, we try different filter sizes and do the training by using CNN again. In the end, the filter size with the best result will be chosen as the final filter size. The RecoG-Net employs an ADAM optimizer for the training purpose and cross-entropy as the loss function during the training. We have determined the architecture of the RecoG-Net through experimentation and chose the architecture that best provided us with higher performance.

Table 1. RecoG -Net Architecture for gate recognition.

Layer	Structure	Output
Conv2d + Relu	32 × 3 × 3	256 × 1047 × 32
Conv2d + Relu	64 × 3 × 3	254 × 1045 × 64
Max pooling2d	3 × 3	84 × 348 × 64
Dropout	0.5	84 × 348 × 64
Conv2d + Relu	32 × 3 × 3	82 × 346 × 32
Conv2d + Relu	64 × 3 × 3	80 × 344 × 64
Max pooling2d	3 × 3	26 × 114 × 64
Dropout	0.5	26 × 114 × 64
Flatten		189,696
Dense + Relu	250	250
Dense + linear	11	11

RecoG-Net is then trained with the training dataset. Then, its performance is measured based on the testing dataset. The predicted outcome of the network is the type of gate, which is classified into 11 different types.

2.3. Training Dataset

The first step towards validating our idea is creating a dataset of all different gates with some specific considered characteristics. The main important feature of the dataset is that it should contain all the essential gates like NAND, NOR, and XOR. Moreover, the input size and the number of inputs for each gate should be considered as other parameters.

In addition to the mentioned criteria, having all the different layers existing in each of the gates like the metal layer is needed. On the other hand, the dataset that was going to be used in the experiments had to be in image format. After thoroughly checking on the existing datasets, we decided to build up a dataset with all the mentioned features. So, to start our experiments, we used GDSII files of 45 nm technology, and extracted different layers of each gate. The generated dataset consists of all the layers of all the gates, with different sizing in image format. Figure 2 shows an example of the constructed dataset of different layers of 2 inputs NAND gate.

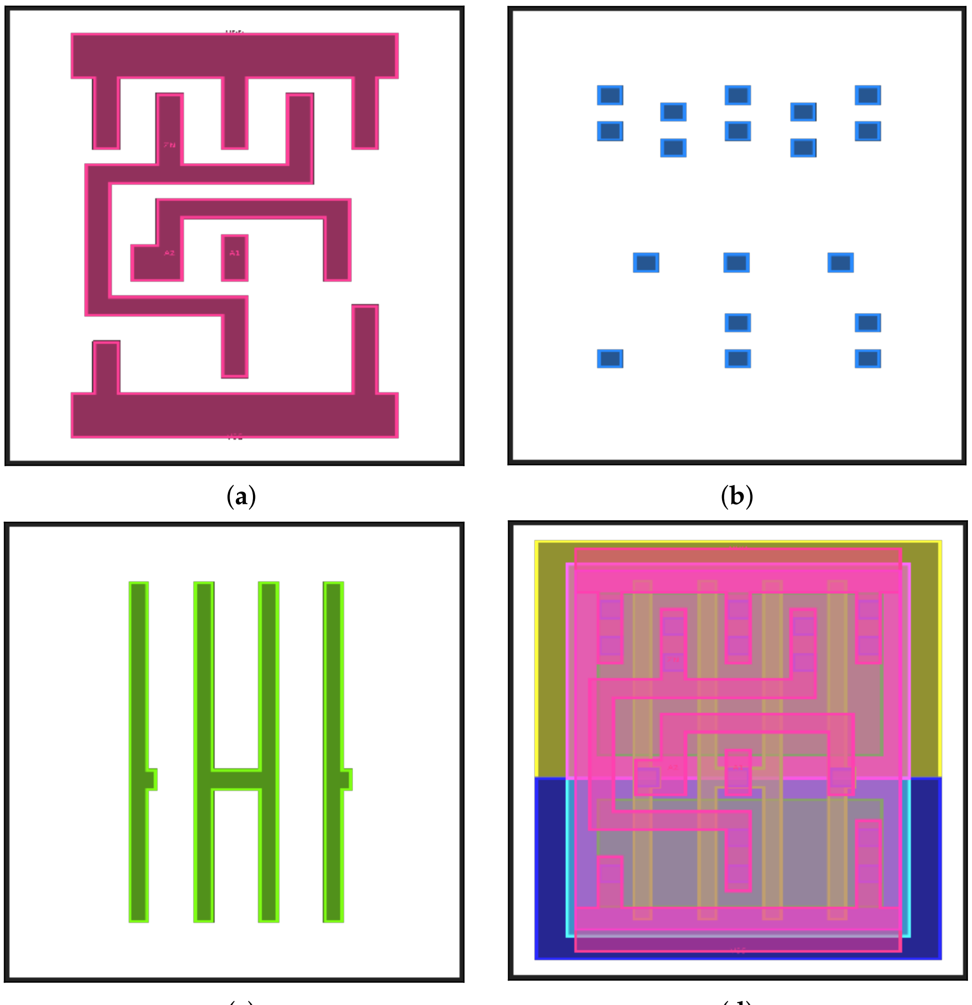
(**a**) (**b**) (**c**) (**d**)

Figure 2. An example of different layers of a 2-input NAND gate. (**a**) Metal 1 layer, (**b**) Contact layer, (**c**) Poly layer, (**d**) All layers.

2.4. Fabrication Impact

As mentioned before, the first contribution of this paper is to find out the type of IC components (gates) using image(s) of one or multiple layers of the gate. To have realistic results, we could not use the generated dataset; since our created dataset was based on the design layout of the gates, all the edges and corners are entirely straight, accurate, and sharp lines. Figure 2 is a valid example of this statement.

Nevertheless, the existing challenge is to distinguish the gates after the physical design and fabrication process. Figure 3 displays an IC and its components after fabrication. Therefore, we made some changes in the created dataset to add the effect of fabrication to increase the resemblance of the dataset to an actual fabricated image of IC.

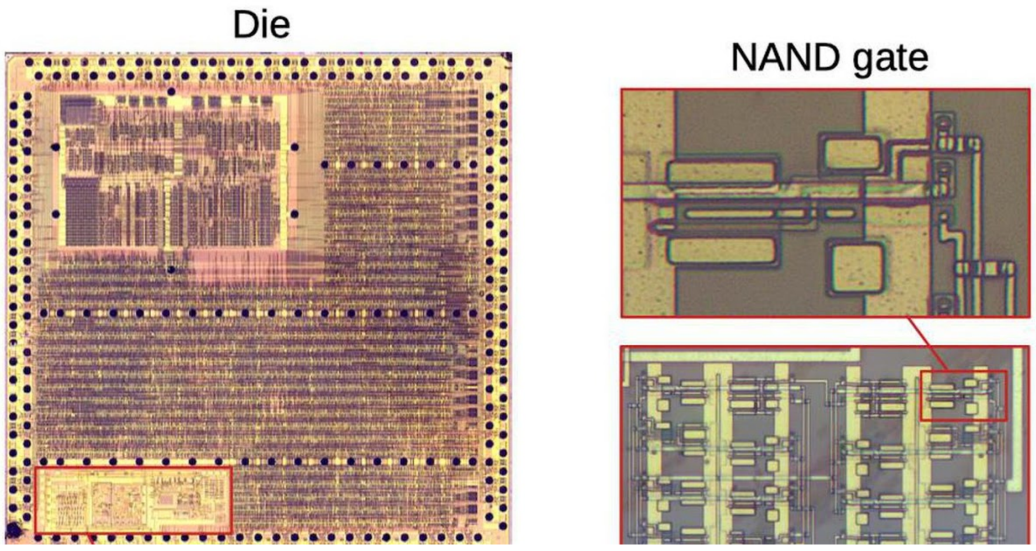

Figure 3. An example of fabricated IC, with a sample of NAND gate inside it.

To solve the problem, we applied an image processing technique named morphing transformation. Morphological transformations are collections of some non-linear operations related to the shape or morphology of an image. In other words, these operations do not rely on the numerical values of pixels in the image; just the relative ordering of the pixels is considered. In this technique, two inputs are required, our original image and kernel or the structuring element. The kernel is the input that is responsible for deciding the nature of the ongoing operation. It is positioned at all possible locations in the image, and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood. Even though different morphological operations exist, no matter what is being used, a morphological operation on an image creates a new image in which the pixel has a non-zero value only if the test is successful at that location in the input image. The two primary operations are Erosion and Dilation, which have been used in this work.

The Erosion technique erodes away the boundaries of the foreground object. In other words, all the pixels near the boundary of an object will be discarded depending on the size of the kernel. So the foreground object shrinks. The dilation technique is just the opposite of the erosion technique. So, this technique will increase the thickness or size of the foreground object. In this paper, to mimic the impact of fabrication, the Closing technique of morphology transformation is also used. The Closing technique helps to close small holes inside the foreground objects. Figure 4 indicates the impact of different morphological transformation techniques on a metal layer of NAND example.

To create a more realistic dataset, first, we applied the Erosion technique. In parallel, the Dilation technique followed by the Closing process was done. Finally, both of the created images were combined to form a fabricated-resembling dataset.

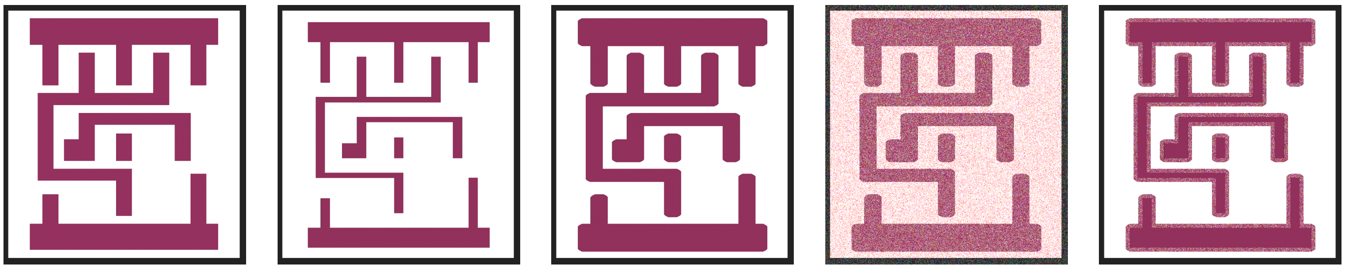
(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 4. Impact of different morphological transformation operations on a metal layer. (**a**) Original, (**b**) Impact of "Erosion", (**c**) Impact of "Dilation", (**d**) Impact of "Close", (**e**) Impact of morphing.

3. Adversarial Perturbation Methodology

We discuss our proposed CAPTIVE technique in this section. The CAPTIVE technique eliminates the possibility of reverse engineering a fabricated IC by inducing specially crafted perturbations named constrained adversarial samples. The perturbed samples include a minimal amount of noise that sometimes can not even be seen by the naked eyes nor affect the recognition capability by naked human eyes but can be misclassified by the ML classifiers.

There exist numerous techniques to inject adversarial perturbations [17,29]. Since the amount of noise added to the image is better to be minimal, we utilize three different perturbations, DeepFool, Square-box, and Jacobian-based Saliency Map (JSMA), which are explained in detail in the Appendix A, to inject noise in the layout images.

Since the chip images will be fabricated, the noises added to the images must follow the DRC rules. The major problem with the added adversarial noise is that they are not DRC-compliant. In Section 3.1, we implemented a method to convert adversarial perturbations to DRC-compliant noises. Later, by using the proposed neural network method in the previous section, we prove that the certainty of the machine learning model in gate recognition would decrease by adding these DRC-compliant noises. Algorithm 1 dedicates our experimental validation setup.

As described in Algorithm 1, the first step is to feed the input GDS II images to the proposed technique. Further, we initialize the noise parameters for the adversarial attack, which will be used to create the noise for the input GDS II images, as described in line 2 of Algorithm 1. Once the noise parameters are determined, the adversarial perturbations will be generated (line 3 of Algorithm 1), as described in Section 3.1. These perturbations under perturbation constraints (derived based on DRC constraints) will be integrated into the original GDS II input image(s) to create the adversarial layout image, as described in line 4 of Algorithm 1. Lastly, the RecoG-Net (Line 5) will test the perturbed images for classification accuracy. If the perturbed images are still classified with higher accuracy, then the steps are repeated.

Algorithm 1: Experimental validation Setup

```
Input: GDSII file
Output: Perturbed GDSII file
1 do
2   | Initialize perturbation parameter;
3   | Generate adversarial perturbation;
4   | Add perturbation Constraints;
5   | Test using RecoG-Net;
6 while High classification accuracy for perturbed data;
```

3.1. Perturbation Generation

Unlike traditional adversarial attacks in machine learning [22,30–35], the perturbation generation needs to be highly constrained, such as the perturbation can only be placed in certain parts of SEM/layout to ensure the DRC and LVS checks are not violated and should be more similar to process variations rather than artificially induced.

To force the noise to be DRC-compliant, the first principal rule is to have some pre-defined shapes that can be fabricated. Hence, as the first step, we used a filter with a specific size and moved this filter across the whole image. The filter size is based on the λ of the technology size. In each step, we compared the original gate image and the image with induced perturbations. If more than half of the pixels in the filter were perturbed, we considered the whole filter as an induced noise. Since the induced noise with the size of the filter should be fabricated at the end, the filter size should not be too small to cause any problem in fabrication, or too big to be meaningless.

The other essential feature is that all the foreground objects in each layer should have a minimum distance based on the technology size. This minimum distance is $2 \times \lambda$. Therefore, the square-shaped induced perturbations can not be added if they have less than $2 \times \lambda$ distance from the existing foreground elements in each layer. To achieve this purpose, we defined a forbidden area outside of each object. Figure 5 indicates the steps to reach a DRC-compliant noise.

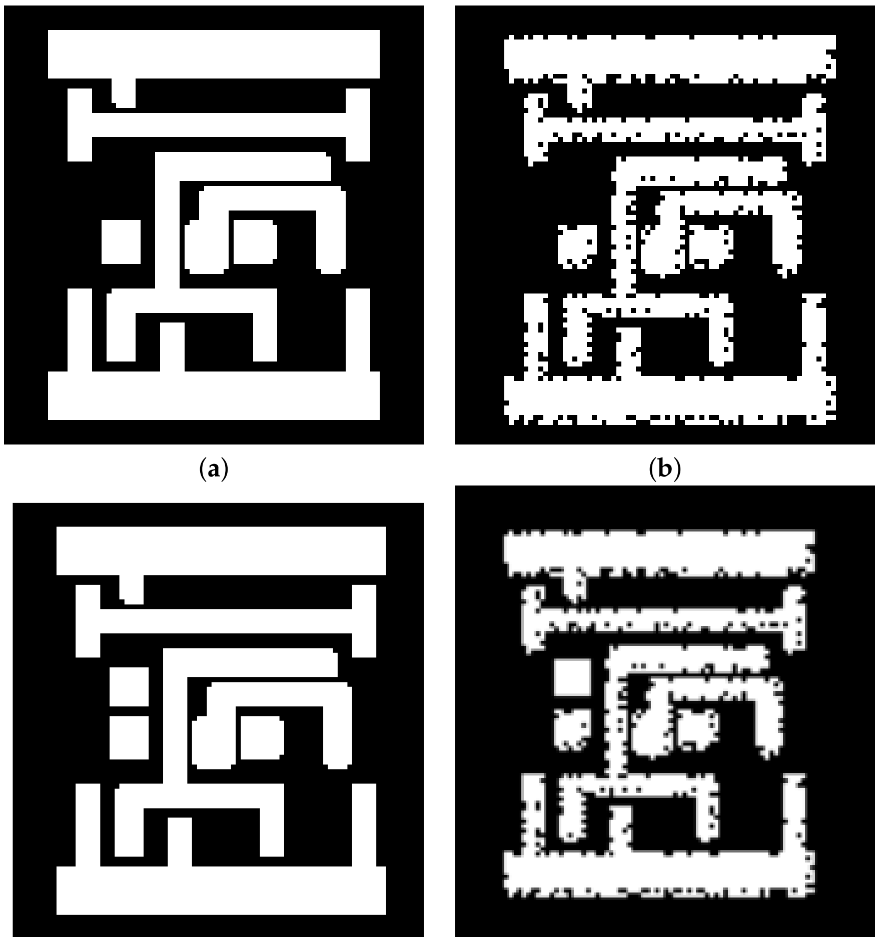
(**a**) (**b**) (**c**) (**d**)

Figure 5. Impact of different morphological transformation operations on a metal layer. (**a**) Original image, (**b**) Morphed image, (**c**) DRC-compliant perturbation added to noise-free image, (**d**) DRC-compliant perturbation added to the morphed image.

3.2. Machine Learning Validation

In Section 2.2, we proposed RecoG-Net, a method for gate recognition. In this method, by using image(s) of different layers of gates as an input of DNN and CNN techniques, we identified the type of gate. The idea is first to add the CAPTIVE DRC-compliant noises. Then, the chip goes to fabrication and after that RecoG-Net will prove the first step of reverse engineering, gate recognition, is not possible.

By adding the introduced DRC-compliant perturbations in the previous section, we can prove that these added noises can decrease the vulnerability of chip reverse engineering. In other words, this method will cause failure in the first step of chip reverse engineering that is gate recognition.

To achieve this purpose, RecoG-Net is trained with a noise-free training dataset. The square-shaped DRC-compliant noises are induced in the testing dataset. Then, the performance of the new perturbed dataset can be compared with the performance of the previous noise-free dataset. In the following section, the results of CAPTIVE are presented.

3.3. DRC and LVS Validation

Design Rule Checker (DRC) checks the layout and verifies if the layout meets all technology-imposed constraints. In comparison to DRC, LVS or Layout Versus Schematic verifies the functionality of the layout. Since there is some induced noise in the GDSII file, it is vital first to pass the DRC rules and then check it with LVS validation tools.

Since the size of the noise introduced as CAPTIVE is based on the technology size and its position is regarding the constraints with the foreground objects, this noise meets the DRC rules. The next step is to check with some LVS software and verify that the CAPTIVE methodology will not change the functionality of the designed gate.

4. Results and Discussion

For the first step of the experiments, we created our own dataset consisting of different layers of the existing gates, such as NAND, NOR, and XOR gates synthesized using standard CMOS 45 nm technology nodes [36]. Using the GDSII file for each gate containing the design for all the layers together, we have extracted an image of 258 by 1049 for each layer. To mimic the impact of fabrication on the generated layer images, we applied different morphological transformation techniques.

4.1. RecoG-Net Results

Since each image consists of just two colors, we converted the images into black and white to get the maximum contrast. After that, by using RecoG-Net, the neural network model introduced in Section 2.2, we tried to recognize the gate based on one/multiple layers of gates. We first shuffle the dataset to get more meaningful results; then, divide it into training and testing data with nearly 80% and 20%, respectively. It is worth mentioning that, due to the shuffling mechanism, we performed each experiment 10 times and reported the average of all experiments as the final result. So, we trained the model using 80% of the data and then tested it with other remaining data points. Table 2 represents the results for using a different layer(s).

Table 2. RecoG-Net recognition accuracy results for different layers.

Layer Name	Contact Layer	Poly Layer	Metal1 Layer	All Layers
Train Accuracy	100%	99%	100%	100%
Test Accuracy	99%	78%	99%	100%

The results proved that RecoG-Net is capable of recognizing gates with 99% accuracy by using just one layer. This layer can be a metal or a contact layer. Hence, we decided to use these two layers to examine the feasibility of adding perturbations to prevent the first step of reverse engineering. As the results indicate, the image consisting of all layers also has 99% accuracy. Since this image contains lots of information and is more complex, we decided to continue two parallel experiments using just the metal and the contact layers separately. The simplicity of our dataset will cause the methodology to be more successful.

4.2. CAPTIVE Results

To achieve the primary purpose of this paper, preventing the chip reverse engineering process, we generated different adversarial datasets using different adversarial techniques. First, we applied three different techniques of JSMA, DeepFool, Square, and their combination on our test dataset separately. Then, some square-shaped DRC-compliant noise generated from different adversarial techniques is added to the test dataset. Finally, the performance of RecoG-Net was tested using the newly adversary-generated test data set.

We did a thorough experiment using different possible perturbation parameters for each type of perturbation. Table 3 presents the best model's performance on different adversarial techniques with different amounts of perturbations that we used. For JSMA, γ and θ are set to 1 and 10, respectively. For DeepFool, the amount of perturbation is considered 0.5. For square-box attack, the best possible parameters are set to $norm = 0$, $max\_iter = 500$, $\epsilon = 0.1$, $p\_init = 0.7$, $nb\_restart = 3$, $batch\_size = 64$. These parameters are similar for both metal and contact layers.

Table 3. Noise-free vs. adversarial vs. DRC-compliant perturbation accuracies with full knowledge of the RecoG-net with 99.8% and 100% noise-less accuracies for metal and contact layers, respectively.

Layer	Perturbation Method	Adversarial Accuracy	DRC-Compliant Accuracy	Improvement
Metal	JSMA	57.5%	63%	36.8%
	DeepFool	50.1%	62.7%	37.1%
	Square-box	32.7%	38.9%	**60.9%**
Contact	JSMA	51%	72.4%	27.6%
	DeepFool	62.5%	67.7%	32.3%
	Square-box	31.9%	46%	**54%**

The results in Table 3 shows that the most effective perturbation method for both metal and contact layers is square-box where for the metal layer the perturbation accuracy dropped to 32.7% and the DRC-compliant accuracy is 38.9%. Also, for the contact layer, the perturbation accuracy and DRC-compliant accuracy are equal to 31.9% and 46%. In addition, the square-box is found to perturb the smallest number of pixels among the other techniques. This makes the square-box perturbation the best candidate for further study to improve the results.

The aforementioned results are performed assuming the designer fully knows the attacker model. In other words, the same network (i.e., RecoG-Net) classifying the gates is used to generate the perturbations. Thus, Table 3 shows the best achievable results. This scenario is rare to happen since usually the attacker and the designer have no relation. Hence, we consider two other scenarios where the attacker has no knowledge of the perturbation network, i.e., consider a black-box attack model.

In the first scenario, the DRC-compliant perturbations are generated using the RecoG-Net model, and the generated images are validated using a different neural network model, whose structure is shown in Table 4. All the parameters used for generating adversaries are the same as before. Table 5 shows the results for two of the networks among many distinct networks that we experimented on. The reason for choosing these two specific networks is their performance on gate recognition when adding DRC-compliant perturbations. Based on the results expressed in Table 5, network "A" has the lowest accuracy in gate recognition compared to network "B", with the highest accuracy in gate recognition. The results in Table 5 illustrate that the most reduction will happen if we apply square-box adversarial perturbation. The possible improvement can be as high as 65.8% or as low as 18% for the metal layer (respectively, 70.4% and 15.5% for the contact layer) regarding the network being used. Even with the lowest improvement in this range, the accuracy of gate recognition is around 80%, which indicates that gate recognition can not be done completely.

Table 4. Architectures for Network "A" and "B" in Table 5.

Network "A"		Network "B"	
Layer	**Structure**	**Layer**	**Structure**
Conv2d + Relu	32 × 3 × 3	Conv2d + Relu	32 × 3 × 3
Conv2d + Relu	32 × 3 × 3	Conv2d + Relu	64 × 3 × 3
Max pooling2d	3 × 3	Conv2d+Relu	64 × 3 × 3
Dropout	0.7	Max pooling2d	3 × 3
Conv2d + Relu	32 × 3 × 3	Dropout	0.5
Conv2d + Relu	32 × 3 × 3	Conv2d + Relu	32 × 3 × 3
Max pooling2d	3 × 3	Conv2d + Relu	64 × 3 × 3
Dropout	0.7	Max pooling2d	3 × 3
Dense + Relu	250	Dropout	0.5
Dense + linear	11	Dense + Relu	250
		Dense + Relu	120
		Dense + linear	11

Table 5. Noise-free vs. adversarial vs. DRC-compliant perturbation accuracies with different recognition networks. Adversary is generated using RecoG-Net.

Layer	**Perturbation Method**	**Network "A"**				**Network "B"**			
		Noise-Free Accuracy	**Adversarial Accuracy**	**DRC-Compliant Accuracy**	**Improvement**	**Noise-Free Accuracy**	**Adversarial Accuracy**	**DRC-Compliant Accuracy**	**Improvement**
Metal	JSMA	99.9%	64%	70.5%	29.4%	99.4%	79%	81.4%	18%
	DeepFool		52.4%	57.1%	42.8%		54%	62.4%	37%
	Square-box		27%	34.2%	**65.7%**		39.4%	43%	56.4%
Contact	JSMA	99.4%	51%	67%	32.4%	100%	67.4%	84.5%	15.5%
	DeepFool		60%	67.4%	32%		68%	75.4%	24.6%
	Square-box		24.5%	29.6%	**69.8%**		30%	36.4%	63.6%

For the second scenario, we have designed several CNNs similar to RecoG-Net and generated the DRC-compliant perturbations using these networks. Afterward, the images were validated using RecoG-Net. The results of one of them are depicted in Table 6. All the parameters used for each of the perturbations are the same as previously mentioned. Table 6 shows that the most reduction in the accuracy happens if the square-box adversarial perturbation is applied, followed by the CAPTIVE method. Clearly, when the perturbation network and attacker network as mismatched, the improvement is dropped by around 40%.

Table 6. Noise-free vs. adversarial vs. DRC-compliant perturbation accuracies with different recognition networks. Adversary is validated using RecoG-Net with 99.7% test accuracy for metal layer and 99.2% for contact layer.

Layer	**Perturbation Method**	**Adversarial Accuracy**	**DRC-Compliant Accuracy**	**Improvement**
Metal	JSMA	59.2%	78.3%	21.4%
	DeepFool	67.4%	76.9%	22.8%
	Square-box	51%	59.4%	**40.3%**
Contact	JSMA	67%	96.2%	3%
	DeepFool	44.1%	75.5%	23.7%
	Square-box	35.6%	71.8%	**27.4%**

4.3. Related Work

Logic camouflaging has been introduced in [37], where the gates are designed to look alike when reverse engineered, but, can function in a different manner. Though able to protect against RE, such techniques lack scalability, can only camouflage a limited number of gates due to EDA constraints, and are shown to be vulnerable to brute-force attacks [38].

On the other hand, ReGDS [39] has introduced a framework that exploits unique relationship-based matching to identify logic gates and thereby, recover the original gate-level netlist. However, the challenge with such frameworks lies in the scalability and accuracy of the matching. One of the first successful trials for IC reverse engineering is Degate tool [14] from academia. Degate is a semi-automated tool to assist in reverse-engineering the digital logic in ICs. Degate tool performs three main steps: template matching to recognize gates where it matches standard cells on the imagery given by a graphical template then via matching followed by wire matching. Then, the SPICE netlist has to be constructed manually.

The work in *Pix2Net* is another powerful tool to extract the schematic and logic gates from SEM images from *MicroNet Solutions Inc.* (Metro Manila, Philippines) [40]. The primer version comes with full circuit extraction, VHDL, or SPICE netlisting with a full schematics library. During the schematics extraction, the Pix2Net performs many steps including layer alignment, stitching, auto cell identification, placement, electrical error checking, and others.

Another commercial and powerful tool is *Circuit Vision* from *TechInsights* (Ottawa, ON, Canada). This tool works from IC-level reverse engineering going up to the system level and functionality. It also supports different types of circuits and chips including analog, digital, MEMS, and others. The tool was tested on many commercial memory chips and major companies including Samsung, SK Hynix, Micron, and Intel. However, the use of commercial and existing tools requires significant expertise in the IC design and major changes to the overall EDA design flow, which is not a pre-requisite for the proposed CAPTIVE.

5. Conclusions and Future Work

This paper demonstrates an experiment about the feasibility of adding DRC-compliant noise to the GDSII file to prevent the first step towards chip reverse engineering, gate recognition. First, we created the dataset containing the images of different layers of different gates using 45 nm technology. Then, we classified them into 11 categories and developed RecoG-Net, a neural network model to do the gate recognition based on the created dataset. As our last and foremost contribution, we have generated DRC-compliant perturbations based on some of the adversarial perturbations. These perturbations are added to the images of different layers. Our experiments report that by adding these DRC-compliant to the images of the metal and contact layers, the performance of gate recognition will drastically drop, which will cause unsuccessful gate recognition as the first step of reverse engineering. We have evaluated the proposed CAPTIVE on different networks, which is different than the network used to generate the adversarial perturbations on the layout. We have crafted a dataset of basic logic gates using CMOS 45 nm technology for evaluation. The proposed CAPTIVE has shown that the recognition of the gates can drop from 100% to 30% after the inclusion of the perturbations without violating the DRC-check rules.

To further improve our methodology, a surrogate gradient perturbation has to be formulated with constraints on the physical design rules. Square-box problem formulation would be a good candidate to start with since it shows the best perturbation accuracy among the investigated methods. Secondly, a combination of different adversarial perturbations needs to be explored and evaluated for enhancing CAPTIVE methodology for robustness and EDA compliance. Thirdly, to overcome the introduced perturbations, an adversary can consider adversarial defenses or strategies. To thwart such scenarios, one can introduce random inclusion of perturbed and non-perturbed gates through the inclusion of both types of gates into the standard cell library, leading to randomness and moving target defense. Such random consideration of gates in the layout can lead to a lower probability or accuracy of reverse engineering by the adversaries.

Author Contributions: Conceptualization, S.M.P.D.; Software, A.H.A.Z. and M.A. Writing—original draft, A.H.A.Z. and M.A.; Writing—review & editing, M.S., S.M.P.D. and M.E.F.; Supervision, S.M.P.D., M.E.F. and F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the US National Science Foundation (NSF) Grant CNS 2213404. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of the US NSF.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors would like to thank Monisha Loganathan for helping to construct the dataset from the technology kit.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The following are summary of the used perturbations methods in CAPTIVE.

Appendix A.1. Jacobian-Based Saliency Map Attack (JSMA)

In contrast to applying noise to every single feature of the input data, [23] proposes an iterative technique to add the perturbation, where the forward derivative of DNN is exploited for adding the perturbations.

Consider a neural network F with input x. If the corresponding output is class j, we represent the model as $F_j(x)$. The main principle of this work is to provide a target t as the output, the probability for $F_t(X)$ must be increased, and simultaneously, the probabilities of $F_j(X)$ for all the other classes i.e., $j \neq t$ have to be decreased, until $t = arg\ max_j F_j(X)$ is achieved. This is accomplished by exploiting the saliency map, as defined below

$$S(X,t)[i] = \begin{cases} 0, \text{if} \frac{\partial F_t(X)}{\partial X_i} < 0 \text{ or} \sum_{j \neq t} \frac{\partial F_j(X)}{\partial X_i} > 0 \\ (\frac{\partial F_t(X)}{\partial X_i}) | \sum_{j \neq t} \frac{\partial F_j(X)}{\partial X_i} |, \text{otherwise} \end{cases} \quad \text{(A1)}$$

For an input feature i starting with the normal input x, we determine the pair of features $\{i, j\}$ that maximizes $S(X,t)[i] + S(X,t)[j]$ and perturb each of the features by a constant offset ϵ. This process is repeated iteratively until the target misclassification is achieved.

Appendix A.2. DeepFool

DeepFool (DF) is an untargeted adversarial attack optimized for L_2 norm, introduced in [22]. DF is an efficient adversarial attack that is capable of producing adversarial samples that highly resemble the original inputs as compared to the aforementioned adversarial samples, especially FGSM and BIM attacks. The principle of the DeepFool attack is to assume neural networks as completely linear with a hyperplane separating each class from another. Based on this assumption, an optimal solution to this simplified problem is derived from constructing adversarial samples. As the neural networks are non-linear in reality, the same process is repeated considering the non-linearity into the model. This process is repeated multiple times for creating adversaries. This process is terminated when an adversarial sample is found, i.e., misclassification happens.

Appendix A.3. Square-Box Attack

In contrast to the other adversarial attacks, which primarily rely on the gradients to insert the perturbations, Square-box attack [24] is a non-gradient attack, which performs a randomized search and inserts square-shaped perturbations, and in each iteration, the perturbation is situated approximately at the boundary of the images. A Square-box attack requires fewer queries for inserting the square pixels compared to other attacks due to the random search and sampling distribution information.

References

1. Chen, J.C.; Rau, H.; Sun, C.J.; Stzeng, H.W.; Chen, C.H. Workflow design and management for IC supply chain. In Proceedings of the International Conference on Networking, Sensing and Control, Okayama, Japan, 26–29 March 2009.
2. Hassan, R.; Kohle, G.; Rafatirad, S.; Homayoun, H.; Dinakarrao, S.M.P. A Cognitive SAT to SAT-Hard Clause Translation-based Logic Obfuscation. In Proceedings of the ACM/EDAA/IEEE Design Automation and Test in Europe, Grenoble, France, 1–5 February 2021.
3. Torrance, R.; James, D. The state-of-the-art in semiconductor reverse engineering. In Proceedings of the 48th Design Automation Conference, San Diego, CA, USA, 5–9 June 2011; pp. 333–338.
4. Akkaya, N.E.C.; Erbagci, B.; Mai, K. Combatting IC counterfeiting using secure chip odometers. In Proceedings of the IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2–6 December 2017.
5. Dhavlle, A. Reverse Engineering of Integrated Circuits: Tools and Techniques. *arXiv* **2022**, arXiv:2208.08689.
6. Quadir, S.E.; Chen, J.; Forte, D.; Asadizanjani, N.; Shahbazmohamadi, S.; Wang, L.; Chandy, J.; Tehranipoor, M. A survey on chip to system reverse engineering. *Acm J. Emerg. Technol. Comput. Syst. (JETC)* **2016**, *13*, 1–34. [CrossRef]
7. Yang, L.; Shi, C.J. FROSTY: a fast hierarchy extractor for industrial CMOS circuits. In Proceedings of the International Conference on Computer Aided Design, San Jose, CA, USA, 9–13 November 2003.
8. Gate-Level Netlist Reverse Engineering Tool Set for Functionality Recovery and Malicious Logic Detection. In *International Symposium for Testing and Failure Analysis*; ASM International: Almere, The Netherlands, 2016.
9. Azriel, L.; Speith, J.; Albartus, N.; Ginosar, R.; Mendelson, A.; Paar, C. A survey of algorithmic methods in IC reverse engineering. *J. Cryptogr. Eng.* **2021**, *11*, 219–315. [CrossRef]
10. Dai, Y.Y.; Braytont, R.K. Circuit recognition with deep learning. In Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Mclean, VA, USA, 1–5 May 2017.
11. Fayyazi, A.; Shababi, S.; Nuzzo, P.; Nazarian, S.; Pedram, M. Deep Learning-Based Circuit Recognition Using Sparse Mapping and Level-Dependent Decaying Sum Circuit Representations. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 25–29 March 2019.
12. Fyrbiak, M.; Strauß, S.; Kison, C.; Wallat, S.; Elson, M.; Rummel, N.; Paar, C. Hardware reverse engineering: Overview and open challenges. In Proceedings of the IEEE International Verification and Security Workshop (IVSW), Thessaloniki, Greece, 3–5 July 2017.
13. Xiao, K.; Forte, D.; Jin, Y.; Karri, R.; Bhunia, S.; Tehranipoor, M. Hardware Trojans: Lessons Learned after One Decade of Research. *ACM Trans. Des. Autom. Electron. Syst.* **2016**, *22*, 1–23. [CrossRef]
14. Torrance, R.; James, D. The state-of-the-art in IC reverse engineering. In *International Workshop on Cryptographic Hardware and Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 363–381.
15. Botero, U.J.; Wilson, R.; Lu, H.; Rahman, M.T.; Mallaiyan, M.A.; Ganji, F.; Asadizanjani, N.; Tehranipoor, M.M.; Woodard, D.L.; Forte, D. Hardware Trust and Assurance through Reverse Engineering: A Tutorial and Outlook from Image Analysis and Machine Learning Perspectives. *J. Emerg. Technol. Comput. Syst.* **2021**, *17*, 1–53. [CrossRef]
16. Holler, M.; Odstrcil, M.; Guizar-Sicairos, M.; Lebugle, M.; Müller, E.; Finizio, S.; Tinti, G.; David, C.; Zusman, J.; Unglaub, W.; et al. Three-dimensional imaging of integrated circuits with macro-to nanoscale zoom. *Nat. Electron.* **2019**, *2*, 464–470. [CrossRef]
17. Ashrafiamiri, M.; Manoj Pudukotai Dinakarrao, S.; Afandizadeh Zargari, A.H.; Seo, M.; Kurdahi, F.; Homayoun, H. R2AD: Randomization and Reconstructor-based Adversarial Defense on Deep Neural Network. In Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD, Canmore, AB, Canada, 2–4 September 2020.
18. Yasaei, R.; Yu, S.Y.; Al Faruque, M.A. GNN4TJ: Graph Neural Networks for Hardware Trojan Detection at Register Transfer Level. In Proceedings of the IEEE/ACM Design Automation and Test in Europe Conference (DATE'21), Grenoble, France, 1–5 February 2021.
19. Yasaei, R.; Yu, S.Y.; Kasaeyan Naeini, E.; Al Faruque, M.A. GNN4IP: Graph Neural Network for Hardware Intellectual Property Piracy Detection. In Proceedings of the IEEE/ACM Design Automation Conference (DAC'21), San Francisco, CA, USA, 5–9 December 2021.
20. Aqajari, S.A.H.; Cao, R.; Naeini, E.K.; Calderon, M.D.; Zheng, K.; Dutt, N.; Liljeberg, P.; Salanterä, S.; Nelson, A.M.; Rahmani, A.M. Pain assessment tool with electrodermal activity for postoperative patients: Method validation study. *JMIR mHealth uHealth* **2021**, *9*, e25258. [CrossRef]
21. Yasaei, R.; Hernandez, F.; Al Faruque, M.A. IoT-CAD: context-aware adaptive anomaly detection in IoT systems through sensor association. In Proceedings of the 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Virtual Event, 2–5 November 2020; pp. 1–9.
22. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2574–2582.
23. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, 21–24 March 2016; pp. 372–387.
24. Andriushchenko, M.; Croce, F.; Flammarion, N.; Hein, M. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 484–501.

25. Lippmann, B.; Werner, M.; Unverricht, N.; Singla, A.; Egger, P.; Dübotzky, A.; Gieser, H.; Rasche, M.; Kellermann, O.; Graeb, H. Integrated Flow for Reverse Engineering of Nanoscale Technologies. In Proceedings of the Asia and South Pacific Design Automation Conference, Tokyo, Japan, 21–24 January 2019.
26. Vijayakumar, A.; Patil, V.C.; Holcomb, D.E.; Paar, C.; Kundu, S. Physical Design Obfuscation of Hardware: A Comprehensive Investigation of Device and Logic-Level Techniques. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 64–77. [CrossRef]
27. Gascón, A.; Subramanyan, P.; Dutertre, B.; Tiwari, A.; Jovanović, D.; Malik, S. Template-based circuit understanding. In Proceedings of the Formal Methods in Computer-Aided Design (FMCAD), Lausanne, Switzerland, 21–24 October 2014.
28. Degate: VLSI-Reverse Engineering of Digital Logic in Integrated Circuits (ICs). Available online: https://www.degate.org/ (accessed on 5 December 2023).
29. Dinakarrao, S.M.P.; Amberkar, S.; Rafatirad, S.; Homayoun, H. Enhancing Adversarial Training towards Robust Machine Learners and its Analysis. In Proceedings of the International Conference on Computer-Aided Design (ICCAD), San Diego, CA, USA, 5–8 November 2018.
30. Biggio, B.; Nelson, B.; Laskov, P. Poisoning Attacks Against Support Vector Machines. In Proceedings of the International Conference on Machine Learning, Edinburgh, UK, 26 June–1 July 2012.
31. Feinman, R.; Curtin, R.R.; Shintre, S.; Gardner, A.B. Detecting Adversarial Samples from Artifacts. *arXiv* **2017**, arXiv:1703.00410.
32. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
33. Lowd, D.; Meek, C. Adversarial Learning. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21–24 August 2005.
34. Matsumoto, T.; Matsumoto, H.; Yamada, K.; Hoshino, S. Impact of Artificial "Gummy" Fingers on Fingerprint Systems. In Proceedings of the Optical Security and Counterfeit Deterrence Techniques IV, San Jose, CA, USA, 23–25 January 2002; Volume 26.
35. Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E.; Roli, F. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In Proceedings of the ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 27–38 November 2017.
36. Cheng, K.L.; Wu, C.C.; Wang, Y.P.; Lin, D.W.; Chu, C.M.; Tarng, Y.Y.; Lu, S.Y.; Yang, S.J.; Hsieh, M.H.; Liu, C.M.; et al. A highly scaled, high performance 45 nm bulk logic CMOS technology with 0.242 μm^2 SRAM cell. In Proceedings of the IEEE International Electron Devices Meeting, Washington, DC, USA, 10–12 December 2007.
37. Yasin, M.; Sinanoglu, O. Transforming between logic locking and IC camouflaging. In Proceedings of the International Design Test Symposium (IDT), Amman, Jordan, 14–16 December 2015.
38. Kolhe, G.; Kamali, H.M.; Naicker, M.; Sheaves, T.D.; Mahmoodi, H.; Sai Manoj, P.D.; Homayoun, H.; Rafatirad, S.; Sasan, A. Security and Complexity Analysis of LUT-based Obfuscation: From Blueprint to Reality. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, 4–7 November 2019.
39. Rajarathnam, R.S.; Lin, Y.; Jin, Y.; Pan, D.Z. ReGDS: A Reverse Engineering Framework from GDSII to Gate-level Netlist. In Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST), San Jose, CA, USA, 7–11 December 2020.
40. Pix2Net Manual. Available online: http://micronetsol.net/html_manual/index.html# (accessed on 12 May 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.