REFICS: A Step Towards Linking Vision with Hardware Assurance [English]

Ronald Wilson / REFICS: A Step Towards Linking Vision with Hardware Assurance / Florida Institute for Cybersecurity Research (FICS)

Problem definition:

Integrated Circuits (IC) and Printed Circuits Boards (PCB) can contain small hidden modifications called hardware Trojans (Figure 1), the goal of hardware assurance is to check the presence of these modifications. In this paper the idea is to use computer vision to detect these malicious modifications. To do this it is necessary to artificially increase the existing data set and use denoising, segmentation, vectorization and deep learning to process the images (Figure 2).

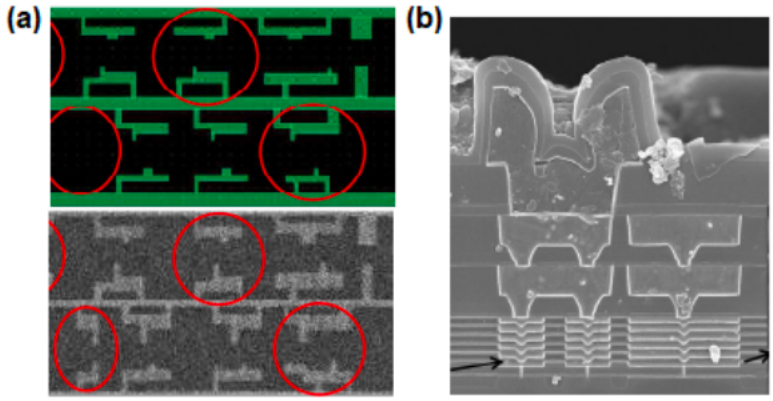

Figure 1. (a) An example of a hardware Trojan [47]. The original layout is on the top and the SEM image of the corresponding location on the IC is on the bottom. (b) Cross-section of an IC captured using SEM imaging indicating multiple layers in its makeup [32].

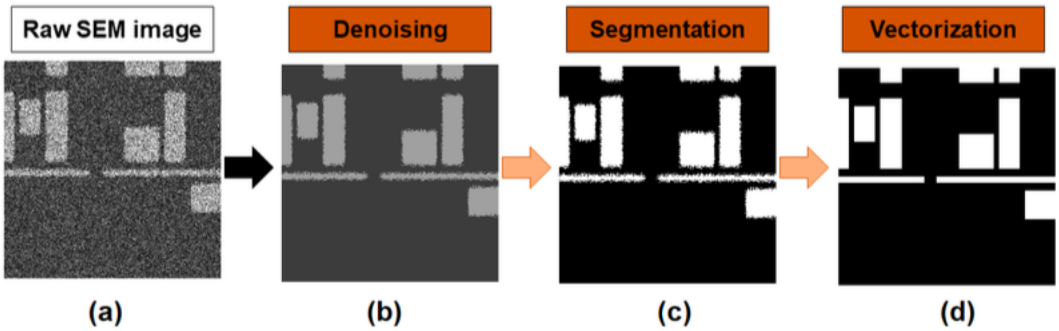

Figure 2. The image processing pipeline for hardware assurance

Motivation

Hardware assurance is an essential process to ensure the good working condition of a hardware device. It mostly uses electron scanned microscopy images (SEM) but it is not well known from the computer vision community. Hardware assurance is really important because IC and PCB are becoming more and more used in our increasing digital world and Trojans in IC and PCB can impact the lifetime of devices or make these devices vulnerable to adversarial attacks. The problem is that the current method to locate these Trojans and to obtain data uses a lot of time and money consuming reverse engineering which is not very efficient. The semi-conductor industry is very protecting towards its technologies, it is why it is hard to obtain a good dataset hence the need to generate one.

Related work:

Denoising:

There are several existing works on how to process noisy SEM images. It includes spatial filtering approaches such as Gaussian, median, curvature, anisotropic diffusion, wavelet, adaptive wiener filter and hysteresis smoothing. You can also find works using high-frequency filtering and DL-based denoising, ML-based approaches.

Segmentation:

Supervised segmentation approaches based on Support Vector Machines (SVM) and Convolutional Neural Network (CNN) were explored. Unsupervised algorithm like K-means Fuzzy C-means, LARSE and Otsu's binarization have also been tested.

Vectorization:

Only simple edge following algorithms have been test in this context.

Deep Learning:

Pix2pix network was utilized to enhance SEM images and CycleGAN was used to transform such images into corner-deformed GT to have more images for comparison. DnCNN is the most used architecture to denoise real-work image photographs.

Idea:

The idea of this paper is to start bridging the gap between the hardware assurance community and the computer vision community by building a data set and benchmark well-known algorithms in order to invite further cross-work between these two communities.

Method:

The first step is to create an artificial data set of SEM images, to do this about 10 000 standard cells from two libraries were used to create the 4 different layers of and IC which are called the doping, polysilicon, contacts and metal layers. Two sets of parameters are important for the synthetization of the images, the first correspond to the imaging settings and the dwelling time per pixel. The other set deals with the noise characteristics.
The second step is to test all the preexisting methods detailed in “related work” on this dataset and see how they perform.

Experiment & Result:

To test the newly generated SEM images a Jensen-Shannon divergence was used to compare the similarity with real images.

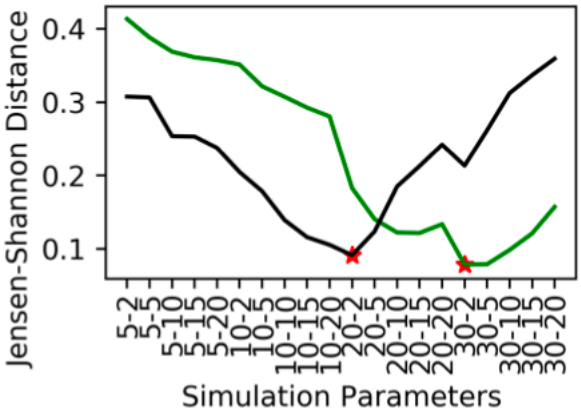

Figure 5. Plot indicating the similarity between the real and synthetic SEM image. Simulation parameter format: Standard deviation of the material-Shot noise parameter. The black and green trends indicate 10 μsec/pixel and 3.2 μsec/pixel dwelling times with the real parameters estimated at 22-2 and 38-2 respectively.

According to the papers the results (Figure 5) showed that the synthetic images were very similar to reals ones. Therefore 800,000 images were created to build the REFICS dataset.

To test all the different existing methods metrics such as intersection-over-union (IoU), structural similarity index measure (SSIM), mean squared error (MSE) and peak signal-to-noise (PSNR) were used to benchmark the different methods. Here are the results summarized in the following tables:

Algorithm	Metal Layer		Doping Layer		Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node	32nm node	90nm node
Denoising Algorithms (Improvement in % for PSNR (↑) / SSIM (↑) over raw SEM image)						
Gaussian fil	8.11 / 22.53	9.46 / 22.24	15.50 / 30.58	15.93 / 32.52	15.84 / 27.99	17.27 / 36.75
Aniso. diff. fil.	1.60 / 45.67	6.37 / 44.79	**26.08** / 72.73	28.64 / 79.16	**29.44 / 62.28**	37.82 / 91.62
Curvature fil.	-28.56 / 29.55	-23.27 / 29.93	-14.65 / 50.02	1.36 / 67.16	18.41 / 52.47	32.46 / 84.95
Median fil.	0.30 / **48.73**	7.01 / 46.83	25.83 / **75.55**	**30.02 / 82.41**	26.82 / 59.91	**42.06 / 94.74**
Adap. Weiner	-27.20 / -29.05	-21.09 / -12.42	-9.73 / 12.05	3.84 / 33.63	9.69 / 30.03	22.81 / 83.45
BM3D	10.20 / 17.99	12.86 / 18.14	11.21 / 22.56	13.54 / 21.22	7.5 / 14.25	12.93 / 19.66
K-SVD	**12.52** / 37.95	**15.32 / 64.73**	23.07 / 49.35	16.00 / 64.27	22.47 / -9.65	23.73 / 65.88
Segmentation Algorithms (SSIM (↑) / IoU (↑) / CC-US (↓) / CC-OS (↓))						
Otsu's thresh.	0.77 / **0.88** / **0.11** / 0.91	**0.79 / 0.91 / 0.13** / 0.69	0.55 / 0.73 / 0.38 / 0.77	0.27 / 0.49 / 0.29 / 0.61	0.40 / 0.52 / 0.64 / 0.69	0.12 / 0.29 / 0.80 / 0.53
Fuzzy C-means	0.75 / 0.86 / **0.11** / 0.91	0.78 / 0.90 / 0.14 / 0.68	0.53 / 0.72 / 0.38 / 0.77	0.27 / 0.49 / 0.30 / 0.60	0.39 / 0.51 / 0.65 / 0.68	0.11 / 0.28 / 0.83 / 0.52
K-means	0.77 / **0.88** / **0.11** / 0.91	**0.79 / 0.91 / 0.13** / 0.69	0.55 / 0.73 / 0.38 / 0.77	0.27 / 0.49 / 0.29 / 0.60	0.40 / 0.52 / 0.64 / 0.69	0.12 / 0.29 / 0.81 / 0.53
HAS	**0.85** / 0.78 / 0.41 / 0.70	0.76 / 0.82 / 0.36 / 0.17	**0.85 / 0.81** / 0.43 / 0.78	0.81 / 0.80 / 0.17 / 0.18	**0.67** / 0.52 / 0.52 / 0.76	**0.56 / 0.46 / 0.33** / 0.60
LASRE	0.75 / 0.70 / 0.15 / **0.14**	0.72 / 0.76 / 0.28 / 0.22	0.78 / 0.79 / **0.09 / 0.20**	0.72 / 0.73 / 0.12 / 0.28	0.46 / **0.58 / 0.30 / 0.38**	0.22 / 0.44 / 0.39 / **0.42**
SVM-10	0.76 / 0.73 / 0.26 / 0.78	0.67 / 0.79 / 0.20 / **0.15**	0.74 / 0.78 / 0.27 / 0.86	**0.85 / 0.85 / 0.05 / 0.11**	0.34 / 0.44 / 0.60 / 0.78	0.32 / 0.37 / 0.61 / 0.47
Deep Learning Algorithms (SSIM (↑) / IoU (↑) / CC-US (↓) / CC-OS (↓))						
DnCNN	0.94 / 0.90 / **0.00** / 0.03	0.92 / 0.92 / **0.00** / 0.10	0.96 / 0.95 / **0.00** / 0.02	0.94 / 0.91 / **0.00** / 0.07	0.83 / 0.67 / **0.00** / 0.48	0.88 / 0.63 / 0.02 / 0.17
CBDNet	**0.96 / 0.94 / 0.00** / 0.03	**0.96 / 0.95** / 0.01 / 0.04	**0.98 / 0.97 / 0.00 / 0.00**	**0.98 / 0.96 / 0.00 / 0.01**	**0.95 / 0.93 / 0.00 / 0.02**	**0.96 / 0.87 / 0.00 / 0.03**
Pix2pix	0.88 / 0.85 / **0.00 / 0.01**	0.72 / 0.70 / 0.01 / 0.04	0.86 / 0.84 / **0.00** / 0.03	0.76 / 0.71 / **0.00 / 0.01**	0.90 / 0.85 / **0.00** / 0.07	0.67 / 0.74 / 0.01 / 0.04
CycleGAN	0.93 / 0.89 / **0.00 / 0.01**	0.78 / 0.74 / 0.01 / **0.02**	0.96 / 0.90 / **0.00** / 0.03	0.95 / 0.72 / **0.00 / 0.01**	0.90 / 0.87 / **0.00** / 0.07	0.91 / 0.62 / 0.02 / 0.04

Table 2. Benchmark of image processing algorithms used on SEM images in hardware assurance. The negative values reported for denoising algorithms indicate degradation in image quality after denoising. For segmentation algorithms, apart from SVM, all other methods are unsupervised. K-means, Fuzzy C-means and HAS use a 5×5 kernel and SVM uses a 10×10 kernel. The highest improvement in metrics for each layer and node technology is highlighted in bold.

Networks	Metal Layer		Doping Layer		Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node	32nm node	90nm node
DnCNN	**0.93 / 0.90 / 0.00** / 0.04	0.91 / 0.91 / **0.00** / 0.09	0.92 / 0.91 / 0.02 / 0.06	**0.96 / 0.93 / 0.00 / 0.02**	0.80 / 0.75 / 0.04 / 0.06	0.83 / 0.41 / **0.00** / 0.53
CBDNet	0.90 / 0.87 / 0.01 / 0.05	**0.94 / 0.94 / 0.00** / 0.05	**0.95 / 0.94** / 0.01 / **0.02**	0.95 / 0.92 / **0.00** / 0.06	**0.83 / 0.80 / 0.01** / 0.03	**0.86 / 0.57** / 0.01 / **0.03**
Pix2pix	0.73 / 0.70 / 0.04 / **0.03**	0.75 / 0.70 / **0.00** / 0.05	0.86 / 0.84 / **0.00** / 0.04	0.71 / 0.65 / **0.00** / 0.06	0.66 / 0.61 / 0.03 / 0.25	0.66 / 0.41 / 0.03 / 0.30
CycleGAN	0.88 / 0.83 / 0.02 / **0.03**	0.79 / 0.72 / **0.00 / 0.02**	0.90 / 0.82 / **0.00** / 0.05	0.91 / 0.70 / 0.01 / 0.09	**0.83** / 0.76 / **0.01 / 0.02**	0.59 / 0.41 / 0.05 / 0.28

Table 3. Cross-node generalizability results. The listed node technology represents the test set with the network trained on the other node. The results are represented as SSIM / IoU / CC-US / CC-OS scores. The highest improvement in metrics is highlighted in bold.

Trained on Metal Layer				
Networks	Tested on Doping Layer		Tested on Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node
DnCNN	0.91 / 0.82 / **0.00 / 0.01**	0.94 / 0.89 / **0.00** / 0.03	0.66 / 0.07 / **0.00** / 0.22	0.76 / 0.01 / **0.00** / 0.02
CBDNet	0.87 / 0.72/ **0.00** / 0.15	**0.95 / 0.90 / 0.00** / 0.05	0.66 / 0.06 / 0.00 / 0.09	0.76 / 0.01 / 0.00 / 0.04
Pix2pix	0.85 / 0.83 / **0.00** / 0.09	0.68 / 0.63 / **0.00** / 0.02	0.51 / 0.53 / 0.12 / 0.18	0.26 / 0.22 / 0.20 / 0.29
CycleGAN	**0.92 / 0.89 / 0.00 / 0.01**	0.78 / 0.73 / **0.00 / 0.01**	**0.75 / 0.69 / 0.09 / 0.17**	0.55 / 0.37 / 0.09 / 0.18
Trained on Doping Layer				
Networks	Tested on Metal Layer		Tested on Polysilicon Layer	
	32nm node	90nm node	32nm node	90nm node
DnCNN	**0.91 / 0.89 / 0.01 / 0.04**	0.85 / 0.87 / 0.03 / 0.32	0.78 / 0.47 / 0.00 / 0.38	0.76 / 0.05 / 0.00 / 0.09
CBDNet	0.88 / 0.82 / **0.01** / 0.13	**0.91 / 0.91 / 0.02** / 0.08	**0.76 / 0.60 / 0.00 / 0.17**	0.78 / 0.17 / 0.00 / 0.14
Pix2pix	0.65 / 0.63 / 0.06 / 0.14	0.69 / 0.67 / 0.04 / **0.06**	0.34 / 0.40 / 0.20 / 0.26	0.31 / 0.23 / 0.19 / 0.25
CycleGAN	0.81 / 0.70 / 0.04 / 0.24	0.79 / 0.57 / **0.02** / 0.20	0.43 / 0.32 / 0.31 / 0.15	0.77 / 0.23 / 0.06 / 0.17
Trained on Polysilicon Layer				
Networks	Tested on Metal Layer		Tested on Doping Layer	
	32nm node	90nm node	32nm node	90nm node
DnCNN	**0.85 / 0.82** / 0.08 / **0.09**	0.83 / 0.79 / 0.08 / 0.12	**0.89 / 0.87** / 0.08 / **0.03**	0.88 / 0.80 / 0.03 / 0.03
CBDNet	0.82 / 0.67 / **0.01** / 0.49	0.90 / **0.87** / 0.04 / 0.12	0.80 / 0.63 / **0.00** / 0.24	**0.95 / 0.87** / 0.04 / 0.12
Pix2pix	0.75 / 0.55 / 0.05 / 0.52	0.64 / 0.55 / 0.05 / 0.24	0.68 / 0.46 / 0.01 / 0.60	0.72 / 0.58 / **0.00** / 0.09
CycleGAN	0.76 / 0.59 / 0.03 / 0.44	**0.91** / 0.72 / **0.02 / 0.04**	0.74 / 0.65 / 0.02 / 0.36	0.93 / 0.69 / **0.00 / 0.02**

Table 4. Cross-layer generalizability results. The results are represented as SSIM / IoU / CC-US / CC-OS scores. The highest improvement in metrics is highlighted in bold.

We can see in these results that these common metrics used to evaluate the methods are not stable and the creation of a novel metric specialized for hardware assurance should be done. We can also see that most approaches are not stable across node technologies and IC layers. Overall, these results show that further studies have to be led.

Conclusion:

This work is just the beginning of a future cross work between hardware assurance and computer vision, due to the lack of data in this field the REFICS dataset was created artificially. Different methods that can be used in hardware assurance were benchmarked and showed the need of deeper studies using computer vision.

I personally thought that the article was quite hard to understand for a person that knows nothing about hardware assurance and that a lot of the conditions of the experiments were not very clear in the paper.

Take home message:

Computer vision is probably the future of Hardware Assurance.