Red Team vs. Blue Team: A Real-World Hardware Trojan Detection Case Study Across Four Modern CMOS Technology Generations

Endres Puschner*, Thorben Moos†, Steffen Becker‡*,
Christian Kison§, Amir Moradi‡, and Christof Paar*
*Max Planck Institute for Security and Privacy, Germany †Université catholique de Louvain, Belgium
‡Ruhr University Bochum, Germany §Bundeskriminalamt, Germany
Email: {endres.puschner, christof.paar}@mpi-sp.org, thorben.moos@uclouvain.be,
{steffen.becker, christian.kison, amir.moradi}@rub.de

***Abstract*—Verifying the absence of maliciously inserted Trojans in Integrated Circuits (ICs) is a crucial task – especially for security-enabled products. Depending on the concrete threat model, different techniques can be applied for this purpose. Assuming that the original IC layout is benign and free of backdoors, the primary security threats are usually identified as the outsourced manufacturing and transportation. To ensure the absence of Trojans in commissioned chips, one straightforward solution is to compare the received semiconductor devices to the design files that were initially submitted to the foundry. Clearly, conducting such a comparison requires advanced laboratory equipment and qualified experts. Nevertheless, the fundamental techniques to detect Trojans which require evident changes to the silicon layout are nowadays well-understood. Despite this, there is a glaring lack of public case studies describing the process in its entirety while making the underlying datasets publicly available. In this work, we aim to improve upon this state of the art by presenting a public and open hardware Trojan detection case study based on four different digital ICs using a Red Team vs. Blue Team approach. Hereby, the Red Team creates small changes acting as surrogates for inserted Trojans in the layouts of 90 nm, 65 nm, 40 nm, and 28 nm ICs. The quest of the Blue Team is to detect all differences between digital layout and manufactured device by means of a GDSII–vs–SEM-image comparison. Can the Blue Team perform this task efficiently? Our results spark optimism for the Trojan seekers and answer common questions about the efficiency of such techniques for relevant IC sizes. Further, they allow to draw conclusions about the impact of technology scaling on the detection performance.**

***Index Terms*—Hardware Trojans, Very Large Scale Integration, GDSII, Integrated Circuits Verification**

1. Introduction

Hardware in the form of digital Integrated Circuits (ICs) forms the basis of all IT systems and frequently serves as the root of trust for security-critical applications. Modern foundries spend billions of dollars in investments to facilitate the rapid advances in semiconductor manufacturing technology seen in recent decades [1], [2]. Consequently, many hardware design houses cannot afford to keep pace and decide to operate fabless instead, i. e., without their own manufacturing facilities. IC production is then outsourced to contract manufacturers (foundries) that offer to fabricate commissioned chips in a portfolio of process technologies. However, these contract manufacturers cannot always be trusted, as they are in the optimal position to intentionally perform stealthy manipulations – i. e., implement hardware Trojans – in the IC designs of their customers [3]. The design houses and foundries involved in the chip making process may be located in places of the world with vastly different cultural, legal and political structures. Thus, it is only reasonable to consider the possibility of adversarial motivations and to be wary of the integrity of critical devices fabricated by untrusted entities. The transport of digital data or manufactured devices between parties is another vulnerable part of the supply chain, as malicious manipulations may also be performed during transit [4]. The most basic example of a malicious hardware Trojan is a kill switch that can disable (parts of) an IC's functionality on demand [5]. Such Trojans can be implemented with a very low overhead [6]. Beyond such comparably simple constructions, many different Trojan designs with varying degrees of sophistication have been proposed in literature – and the possibilities seem almost endless. We review the relevant state of the art in Section 5. We conclude that fabless design houses are in dire need of techniques to verify that commissioned chips are produced and delivered exactly as ordered, without any intentional or unintentional modifications. Such means should also be in the interest of trustworthy foundries and shipping companies to increase their customer's confidence in the honesty of their business model. In this paper, we therefore address the following research question in a holistic manner:

> *How efficiently can we detect functional hardware Trojans[1] in full-sized ICs manufactured in progressively smaller CMOS technologies?*

To answer this research question, we analyze the detectability of tiny and hidden silicon modifications in four differ-

1. That is, Trojans that are physically realized by adding or modifying gates according to the taxonomy of Karri et al. [7].

ent digital IC prototypes designed for research purposes and manufactured by an external foundry overseas. The four chips correspond to CMOS technology nodes $90\,\mathrm{nm}$, $65\,\mathrm{nm}$, $40\,\mathrm{nm}$, and $28\,\mathrm{nm}$ and their digital functionality is realized from 450 000 up to 1 500 000 connected standard cells. We have chosen a scientific setting for evaluating the detectability of the minimal manipulations that aspires to avoid biases as much as possible. A bias naturally occurs, for example, if the evaluation of the detection performance and the insertion of the hidden modifications is done by the same individuals or team. Pretending to search for changes while already knowing where to find them is clearly a suboptimal approach to an independent and realistic assessment. Thus, for this work we have chosen to apply the *Red Team* vs. *Blue Team* concept, where Trojan adversaries and Trojan seekers operate strictly separated from each other [6]. In this context, a subgroup of the researchers involved acts as an untrusted, malicious foundry that supposedly inserts a hardware Trojan into the mask layout before manufacturing. This group is called the *Red Team*. To emulate a malicious modification without actually being in position of the foundry, the Red Team manipulates the chip layout instead, i. e., the Graphic Design System II (GDSII) file, after production of the chips. In this way, a difference between physical device and digital layout is created. To be more precise, the Red Team emulates two different types of manipulations at the GDSII–level, which, according to [8], represent realistic Trojan injection capabilities of a malicious foundry:

1) Replacing filler cells, i. e., unused spaces, by functional standard cells to realize the Trojan.
2) Substituting existing cells with cells of a different functionality.

To emulate the replacement of a filler cell by a functional Trojan logic cell in the physical chip, the Red Team simply has to perform the opposite operation in the GDSII file, namely replacing a functional logic cell by a filler cell. To an outside observer, it then looks as if logic has been added to the physical chip compared to its digital layout. We focus on the replacement of filler cells in the first type of modification because of a convincing Trojan insertion strategy first presented at ISCAS 2021 [9] and subsequently demonstrated on real chips in 2022 [10], [11], [12]. The cited works show how effortlessly Trojans can be inserted on the foundry's side with existing design tools by means of an Engineering Change Order (ECO) flow. This attack requires minimal knowledge about the chip to be trojanized and the original design remains untouched as the Trojan functionality is added. The second type of modification (replacing logic by logic) is the more stealthy scenario where the Trojan logic is not added in addition to the existing design, but the existing logic's functionality is changed to realize the Trojan. Clearly, this second scenario requires an adversary who has obtained an in-depth understanding of the targeted chip design.

Working in strict separation from the Red Team are the researchers of the *Blue Team*, which resembles an external analysis laboratory commissioned by an IC design house. The goal of the Blue Team is to uncover all the manipulations made by the Red Team with as little information as possible and, most importantly, without any a-priori knowledge of the IC design under inspection. We assure that the Blue Team has not been involved in the original design of the analyzed chips, has never obtained the Trojan-free GDSII design files and has not been given any hints from the Red Team where to find the hidden modifications. Indeed, apart from a few physical samples (naked) of the manufactured dies of each supposedly trojanized chip, the Blue Team receives merely a stripped-down version of the GDSII design files. As this work focuses on functional Trojans which are realized by adding or replacing library cells, the Blue Team requires only the cell layer of the standard cell area that contains nothing but a labeled bounding box for each instance of a filler or functional standard cell on the chip. To be precise, the GDSII design files received by the Blue Team are free of cell layouts and routing information, such as metal or via layers. This is comfortable for design houses that may not be willing to disclose full details of their digital design to another third-party contractor like an external analysis laboratory.

To detect the manipulations, the Blue Team first captures high-resolution images of the backside of the naked dies using a Scanning Electron Microscope (SEM) and then superimposes these images with the stripped-down GDSII files to check each individual cell on the device for its conformity with the digital layout. To perform the comparison, template matching and via detection mechanisms are employed. We provide a more detailed description of the threat model considered in this work, including its limitations and an ethical discussion of potential biases in our process and its integrity in Section 2.

1.1. Our Contribution

To the best of our knowledge, this is the first work in the public literature to perform and document a comprehensive hardware Trojan detection case study on full ICs over multiple CMOS technology generations. Despite the fact that the fundamental techniques to detect hardware Trojans that require evident changes to the silicon layout (e. g., adding or replacing logic cells) are nowadays well-understood, there exist only few public case studies describing and documenting the process in its entirety. Furthermore, to our knowledge, none of them make the underlying datasets publicly available. So far, previous works could only demonstrate the basic detectability of cell manipulations, albeit on small devices (e. g., 15 000 cells [13] or 40 000 cells [14]) and in outdated CMOS technology generations (e. g., $130\,\mathrm{nm}$ [13] or $180\,\mathrm{nm}$ [14]). For a detailed discussion of related work, see Section 5. The limiting factors for academic institutions to produce more in-depth results in this field include the high cost and effort associated with semiconductor prototyping and analysis, the long production cycles, the limited access to advanced technologies and laboratory equipment, but also technicalities like non-disclosure agreements, copyrights and intellectual property considerations.

In this work, we overcome these obstacles and present a

public and open hardware Trojan detection case study based on GDSII-vs-SEM-image comparisons using four different digital ICs (90 nm, 65 nm, 40 nm and 28 nm) employing a Red Team vs. Blue Team approach. Our results demonstrate that efficient detection of malicious manipulations inserted according to our threat model is possible on all four devices, although not with perfect accuracy. In total we have to tolerate a few hundred false positives (out of more than 3 million cells on all chips combined) that are primarily caused by imperfect SEM-images and require manual inspection by an expert. Additionally, our detection missed a total of three cell replacements, i. e., false negatives, related to instances where the original cell and its replacement appear extremely similar on SEM-images. We observe that the detection performance, while not heavily, suffers noticeably from the down-scaling of the physical feature size. In fact, the majority of false positives and all false negatives are related to finding cell replacements on the 28 nm chip. Our analysis suggests that this is not only related to the difficulty of scanning the nanometer-scale structures with sufficient quality for a meaningful comparison. Instead, it is also related to the higher complexity of gate libraries in smaller technology generations and the larger number of similar-looking standard cells. Also, since our chip imaging was not performed in a cleanroom, it is inevitable that tiny pieces of dust and debris are included in the images and obscure some cell fragments. Clearly, dust and debris of the same size conceal more information in the advanced technology generations, simply because the individual cells are smaller. To ensure that our results are reproducible and can be verified by independent researchers, we publish the complete set of SEM-images and provide all our source code as an easy-to-use open-source toolkit to combine images and design data and apply our Trojan detection algorithms. We also make the modified stripped-down GDSII design files available (see Section 4.6). Thus, researchers around the world can try to be (or to beat) the Blue Team. We believe that our extensive dataset will be a fertile ground for further research in this area.

2. Threat Model and Implications

In this section, we review the threat model that underlies our scientific approach for the detection of hardware Trojans. Afterwards, we discuss the validity of our Trojan surrogates and explain which types of Trojans are not covered by our detection methods. We conclude the section with an ethical discussion of potential biases remaining in the procedure and make assertions about the integrity of our process.

2.1. Threat Model

Adversary models in the world of hardware Trojan insertion, prevention and detection come in all shapes and forms. Pre-manufacturing threats include, amongst others, untrusted employees, subverted design tools and malicious third-party IP cores [15]. With most design houses operating fabless these days, there are not only internal security risks, but external ones as well. Delivery of the digital design files, the manufacturing period at an untrusted foundry, and transportation of the produced devices by a third-party contractor are all attack vectors that must be considered.

In this work, we assume that the finished chip design leaving the IC design house is free of Trojans or backdoors and therefore not targeted or affected by internal threats like untrusted employees, subverted design tools, or malicious third-party IP cores. This is conceptually justified by the degree of control over internal threats compared to external ones. In our threat model, we only consider external security risks, in particular outsourced manufacturing and transportation, which are indeed associated with the greatest danger of malicious subversion. We focus primarily on Trojan insertion methods that seem plausible in the real world, and pay somewhat less attention to extremely sophisticated Trojans that require tremendous effort, a-priori knowledge, and skill. As mentioned in Section 1, we focus in a first step on foundry-side attacks that are performed via an ECO flow [9], [10], [11], [12]. These Trojan insertions are among the most realistic that have been demonstrated in the literature. The technique relies on removing filler cells, unused spaces, from the original design and adding the Trojan logic instead. This insertion mechanism is more realistic than other techniques for two simple reasons. 1) The Trojan design can be automatically placed through a professional synthesis and place and route flow with proven timing closure mechanisms. Thus, the Trojan is almost guaranteed to work. 2) The original design, its functionality and timing are almost entirely unaffected by the additional logic (see [9] for a discussion), as no functional cell or wire of the original design is modified. Yet, since filler cells can be distinguished with high probability from logic or sequential cells on SEM-images, this type of Trojan is comparatively easy to detect when scanning the polysilicon layer of the device. Thus, we also consider a second type of adversary in our threat model, which is able to modify cells of the original design and, in particular, replace them with other cells with a different functionality. Performing such manipulations by hand to realize an effective hardware Trojan usually requires a high level of knowledge about the targeted design so as not to break the original functionality or cause behavioral changes that can be detected from the outside. Among identically-sized cells there are always some which appear very similar on SEM-images. Therefore, detection in these cases is significantly more challenging than recognition of replaced fillers.

In summary, we consider all adversaries who can replace functional standard cells or fillers with other functional standard cells or fillers after the digital layout has left the fabless design house and before the physical devices are received.

2.2. Limitations

Our threat model does not capture several types of hardware Trojans which have been discussed in academic literature. As detailed above, internal threats are not considered in this work, so any Trojan inserted into the chip design before it leaves the design house is out of scope. Our technique is

based on comparing a benign design file to the received device. If digital layout and physical chip are identical, but both are trojanized, our method is essentially useless. Yet there are also Trojans which can be inserted during transport or manufacturing at an untrusted foundry that are not covered by our threat model and therefore may not always be detectable. We argue, however, that such Trojans which require less evident changes to the silicon layout, correspond to rather complex and difficult insertion scenarios requiring significant a-priori knowledge about the targeted design. These include hardware Trojans that are inserted merely by changing the routing on a chip, while all cells and their placement remain untouched. While there are techniques to image the routing with a SEM by delayering the chip step-by-step from the front side, this process is time-consuming and error-prone [16]. Thus, we decided to not consider such techniques in this work. New non-invasive scanning methods based on X-Rays [17] seem more promising for the future than the lengthy process of delayering and imaging the chip. These non-invasive techniques are potentially able to scan all metal layers and provide a 3D-image of the entire routing without destroying the device, but the research on this subject is still at an early stage. A third type of Trojan, which is not covered by our threat model and may not be easy to detect with our approach, is based on dopant-level manipulations [18]. Changing the doping of certain transistors to modify their behavior is a potentially very stealthy way of introducing hardware Trojans. However, it has been demonstrated that at least in principle such manipulations can be made visible in SEM-images [19]. We make no claims about whether such detection would be possible in our images and with our setup. Finally, there is a similar Trojan which is based on subtle manipulations at the sub-transistor level to modify the behavior of transistors [20]. The authors list multiple techniques to change the timing of certain paths in an implementation to build a Trojan. These include changing the drive strength, threshold voltage or gate length of transistors in logic cells. Such changes, while potentially rather unobtrusive, usually require scaling of certain visible dimensions or features of the transistors like their width, channel length or gate oxide thickness. Thus, they usually lead to a (slightly) different appearance of the affected cells on SEM images, but again, we make no claim as to whether these differences would be sufficient for effective detection of manipulations with our approach.

2.3. Validity of Trojan Surrogates

We argue that our Trojan surrogates, realized through random cell replacements, are indeed a valid representation of actual inserted functional Trojans when it comes to evaluating the likelihood of their detection, even conceptually close to a lower bound. First of all, we would like to emphasize that most realistic functional hardware Trojans, especially those that have been demonstrated in practice, require adding or adapting multiple cells (often hundreds). Typically, these malicious modifications would appear clustered in one specific region of the chip, making them easier to detect for human analysts. An insertion via ECO into a design that already has a high utilization (ratio of functional cell area to filler cell area) might appear more scattered, but there will still be a strong spatial relationship between the Trojan cells to enable their successful routing. However, it is also possible, albeit challenging, to build an effective functional Trojan by exchanging only a single cell in a target design. These Trojans, which are typically limited in their capabilities, indeed constitute the most stealthy insertion of a functional Trojan, which, by definition, always requires the addition, removal or replacement of at least one single cell (i. e., either a change from filler to standard cell, vice versa, or from standard to standard cell). Thus, in principle, each of the random insertions we consider in our work can be viewed as a functional Trojan of minimal size that stands on its own. As our insertions are realized independently, at random locations on the chips and without any spatial relationship to each other, we believe it is legitimate to consider them as separate entities. In consequence, the detection of any actual inserted functional Trojan may be at most as difficult as the most inconspicuous random single-cell replacement of the type that we consider in this work.

As a cautionary note, however, we would like to mention that it is entirely possible that our randomly selected cell replacements do not happen to cover the single, most stealthy cell replacement possible on each chip. Thus, theoretically, a very sophisticated adversary with knowledge of the underlying cell library and the appearance of each cell type on backside SEM images could indeed perform a cell replacement that is slightly more stealthy than the most stealthy ones considered in this study. Yet, identifying cell types and instances suitable for such clandestine replacements and building an effective Trojan on that basis seems a very advanced, almost impractical, insertion strategy. We also highlight that we did not observe a single pair of cells (with different functionality) on any of the chips, which is indistinguishable to the human eye on the SEM images.

2.4. Potential Biases, Integrity of the Process

In the following, we describe the interaction between Red Team and Blue Team and pinpoint biases that could potentially originate from this very communication. We declare that we have kept the standards for the integrity of our process as high as possible.

First of all, it needs to be mentioned that the Blue Team was informed that each combination of chip and design file indeed contains differences, i. e., surrogates for inserted hardware Trojans, which were supposed to be found. Thus, the Blue Team was aware of the presence of hardware Trojans before the detection even started, which would not be the case in the real world. Theoretically, this might create a bias where the Blue Team devotes more effort into finding the Trojans on these chips than they normally would. However, comparing the detection performance of our techniques across all four chips was too important for us to keep any of them without manipulation.

The Blue Team also received information on how many cells were modified per chip and how they were divided between the two categories (four filler cell and six standard cell

replacements). We decided to keep these numbers constant among all four ICs in order to ensure optimal comparability. Theoretically, this might create a bias where the Blue Team continues the search until all manipulations have been found. However, our detection is based on checking each individual cell on the physical device for conformity with the digital layout. In that regard, the total number of suspicious cells is not critical, as successful detection of a manipulation can be reported once a single modified cell has been detected with certainty. In our analysis, once a good set of parameters for the detection had been found, almost all modifications have been identified by the algorithm at the same time. Instances where this was not the case are explicitly described in Section 4.

Apart from the information about the existence of Trojans and the number of affected cells, no further information was shared with the Blue Team by the Red Team. In detail, we assert that the Blue Team has *not received any information about the location, size, name, type, logic functionality, drive strength, voltage threshold, rotation, mirroring, or other related information about the manipulated cells*. Further, the modifications have been scattered over the entire standard cell area rather than clustered in a specific region to complicate the detection and not allow the Blue Team to deduce the location of one modification from another. In summary, we assert that all detection results reported in Section 4 Table 2 have been obtained without any help or intervention from the Red Team.

3. Methods

In this section, we provide an overview of the sequence of steps in our experimental methodology. First, we introduce the target Application-Specific Integrated Circuits (ASICs) which form the basis for our analysis. We then describe how the Red Team creates the differences between digital layout and the physical device to emulate the Trojan insertion, and give details about the type and number of manipulations per device. We then explain in-depth the process followed by the Blue Team to determine whether hardware Trojans can be detected. The process involves, among other steps, the preparation of the physical samples for imaging, the imaging process using a SEM, and the detection mechanisms to compare images with layout files to find differences.

3.1. Target ASICs

The four IC prototypes used to analyze our Trojan detection capabilities were developed for independent research purposes in the field of cryptographic hardware.[2] The chips contain hardware co-processors for common block ciphers hardened against so-called side-channel and fault injection attacks.[3] The ASICs are realized only from digital IO cells and digital standard cells and contain no IP blocks, no memories and no analog components. Thus, they are perfectly suited to test how efficiently tiny modifications hidden in a large pile of standard cell instances can be detected. It is important to mention that all four chips are manufactured in planar bulk CMOS technology. Figure 1 depicts the layouts of our ICs in the top row.

2. More detailed information on the prototypes and their previous utilization for hardware security research can be found in [21].

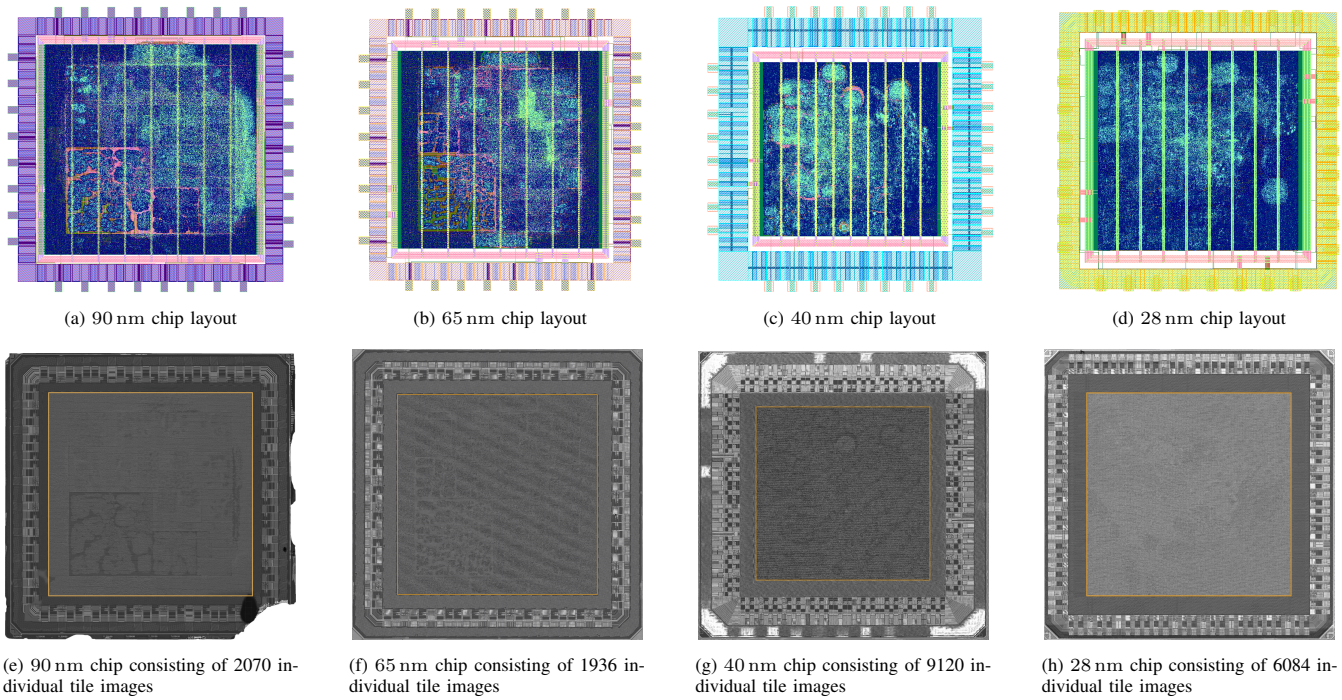
3. For brevity, we do not introduce these terms here. The topic is mostly unrelated to this work.

3.2. RED TEAM Target Selection and Manipulation

The Red Team's task is to manipulate the design files to appear as if the fabricated chips have been the victim of a malicious modification by a third party such as a foundry or a transportation contractor. We call this *emulating* a Trojan insertion. As the Red Team is not actually a malicious foundry, it manipulates the chip design files after the chips have been manufactured instead of changing the layout or masks before manufacturing to create an evident difference between digital design file and physical device. The resulting difference is indistinguishable from an actual foundry-side attack, when the Red Team simply performs the opposite operation in the GDSII file than one would expect from an untrusted foundry. That means, for example, that functional logical or sequential cells in the design file are replaced by filler cells, which are essentially blank dummy cells with no functionality. This way, it appears as if the physical device has more functional cells than it is supposed to have (according to the supposedly benign design file), which is a clear indication of a hardware Trojan insertion.

3.2.1. Manipulation A: Replaced Filler Cell. The first type of manipulation is based on the exact idea we just described. The Red Team emulates the removal of filler cells and their replacement with additional functionality in a final chip design. This Trojan insertion scenario can even be assisted and automated by commercial Electronic Design Automation tools due to the Engineering Change Order flow [9]. This flow has been established for benign purposes, namely for cases where for legitimate reasons small changes or additions have to be introduced into final designs of chips without repeating all design phases from scratch. Yet, it can easily be abused by malicious entities to add Trojan logic to someone else's design. Hence, this foundry-side attack is one of the most realistic insertion settings. In our case the Red Team has manually emulated the replacement of *four* filler cells of different sizes (4, 8, 16 and 32 units) by functional standard cells on each of the chips. The modifications have been performed in the GDSII design file using the open source tool *Klayout 0.25.7 Editor*. The locations have been selected at random, scattered all over the chips' standard cell areas.

3.2.2. Manipulation B: Replaced Functional Cell. The second type of manipulation involves replacing functional standard cells with other functional standard cells. The idea behind this type of modification is to significantly raise the detection difficulty, since the differences between two types of equally-sized standard cells may be small. In fact, the distinguishability between different functional standard cells is usually much lower than between a functional standard cell and a filler cell. Specifically, the Red Team selected *six* standard cells at random locations on each chip and replaced them with equally-sized but different standard cells.



(a) 90 nm chip layout (b) 65 nm chip layout (c) 40 nm chip layout (d) 28 nm chip layout

(e) 90 nm chip consisting of 2070 individual tile images (f) 65 nm chip consisting of 1936 individual tile images (g) 40 nm chip consisting of 9120 individual tile images (h) 28 nm chip consisting of 6084 individual tile images

Figure 1. Top: Layouts of the four evaluated ASICs. Bottom: Fused SEM images of the backside of the chips. The orange rectangles delimit the respective standard cell area. The defects seen at some of the edges are due to etching and do not affect detection as we only focus on the standard cell area.

3.3. BLUE TEAM Sample Preparation and Imaging

The Blue Team receives the modified design files and a few physical samples (naked dies) of each supposedly trojanized chip. To create imagery of the chips' standard cells, i. e., the polysilicon layer, the Blue Team prepares the samples from the backside. Thus, several µm – usually 150 µm or more – of silicon need to be removed from the bulk in a number of consecutive steps. For this purpose, the chip is first mechanically milled in a rough pass to remove most of the silicon. This process thins the bulk as close to the backside as possible without damaging the chip or bonding wires. Here, 10 to 15 µm Remaining Silicon Thickness (RST) are feasible without causing damage to the structures. For our particular purpose, a polishing step is not required as the remaining part is wet etched. In fact, a rough surface should significantly speed up the etching process due to the enhanced contact surface with the chemicals in the next step. The remaining silicon is etched with wet chemicals, e. g., tetramethylammonium hydroxide (TMAH), potassium hydroxide (KOH) or choline hydroxide (CH) [22]. Since the Blue Team already had experience with the use of choline hydroxide, it was decided to etch with this substance. The ratio of etch rates between silicon (Si) and Silicon dioxide (SiO_2) is high enough to induce a highly selective etch process that slows down significantly at the SiO_2 barrier in a low temperature environment. In this way, the silicon backside of a chip can be etched and thinned in only about one to two hours – assuming the appropriate experience and a well-equipped laboratory.

The chips prepared in this manner can then be imaged in a resolution suitable for each technology node. Each chip is then digitized by tiling images, i. e., by creating a mosaic of a large image. Each captured image tile is a 4096 by 4096 pixel sized 8-bit gray scale image. A complete chip consists of up to 9120 image tiles (see Figures 1e, 1f, 1g, and 1h) with an overlap of 10% to adjacent images to facilitate stitching. The standard cell area of the chips relevant for our analyses is already covered by up to 3364 tile images. After acquisition, the images are stitched together by correlating the overlapping areas. This can be done with well-established software for stitching microscopy images, e. g., MIST [23] or BigStitcher [24] which both are packages in the Fiji/ImageJ suite [25], [26]. The resulting fused images of the chips are shown in the bottom row of Figure 1.

Costs and Required Expertise. The cost of consumables for the IC chip backside preparation is negligible compared to the procurement cost of a SEM and a milling machine. For our experiments, we did not have special laboratory requirements – in particular, we did not utilize a cleanroom – and used inexpensive fixtures (platinum plates) to handle the ICs. However, a modern SEM is required depending on the targeted IC technology sizes. We used a *FEI Helios G4 Dual Beam* Focused Ion Beam (FIB). We only utilized the device's SEM capacity and did not make use of the FIB. The combined costs of a modern SEM and milling machine such as those used for our experiments are around 500 000 to 700 000 US dollars. If not available, a SEM or milling machine can be rented for the expected duration, even with an experienced operator. Our wet chemical and SEM operators are professionals with more than ten years of experience, but the steps can be taught to novices as the chemical process is very forgiving and almost self-stopping.

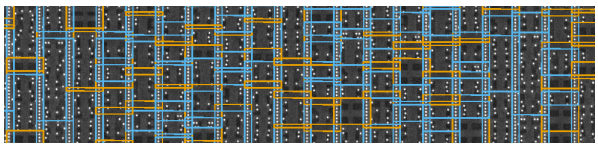


Figure 2. Bounding Boxes over parts of a single tile of a chip. The orange borders mark filler cells, blue borders mark functional standard cells.

Also, the image tiling inside the SEM is an automated process. Nevertheless, the initial SEM setup remains challenging and requires experience and the incorporation of continuous feedback from the image processing steps.

3.4. BLUE TEAM Mapping Design Files to Images

The first step in detecting manipulated standard cells in a stitched microscope image of a chip is to overlay the microscope imagery and the GDSII design file, which contains a bounding box for each cell, as accurately as possible. Figure 2 shows a segment on which images and design files are overlaid. Then an image of each cell instance can be created, along with the specific functionality defined in the design file. Based on this information, further algorithms can detect potentially manipulated cells.

3.4.1. Perspective Translation Matrix. To overlay images and design files, the coordinate system of the microscope image has to be translated to the coordinate system of the GDSII or vice versa. Only if the coordinates can be translated with sufficient accuracy, the further analysis steps will yield useful results. One obvious fact to consider is that the backside images must be mirrored on one axis to match the GDSII. This is implicitly the case as the coordinate system of the GDSII format is inverted on one axis in comparison to computer image coordinates. In addition, the SEM images may need to be rotated perpendicularly (by 90/180/270 degrees). Significantly more effort is required, however, to fix (even minor) stitching errors or slight stretching and rotation. To correct these errors, we need to create a perspective translation matrix based on the four edges of the chip structure with their respective stitched global image coordinates. Further variations in the stitching can be compensated for by tile-level corrections, which can also be interpolated to avoid having to correct each individual stitching error manually.

3.4.2. Power Line Detector. Additional structural features such as power lines can also be utilized to fix at least one dimension to a known baseline. As such, when correcting cell cutouts, they can be used as guiding lines that delimit the top and bottom of the cell instance. Power lines are distinct visible features on the tile images and their main property is that they run straight from one edge of the standard cell area to the other. They are always located between two rows of cells and therefore maintain the same distance from each other across the entire chip. Power lines basically appear as tracks, but depending on chip technology and imaging, they can also consist of many adjacent vias to efficiently route supply voltage between chip layers.

To determine the positions of the power lines, straight edges through the entire standard cell area have to be detected. Technically, a slight directed blur is applied in the direction of the cell rows to prepare the tile image. Then the Sobel edge detector [27] is applied to the blurred image, followed by eroding and binary thresholding, and a second directed blur to further smooth the result. At this stage, the only visible lines across the entire image dimension are power lines, which can be formalized by applying the Hough transform [28]. In the end, the resulting lines are checked for plausibility: Only power lines that are within the average distance to other power lines are considered. If a power line is not detected for any reason, it can be reliably inserted with the estimated average distance to adjacent power lines.

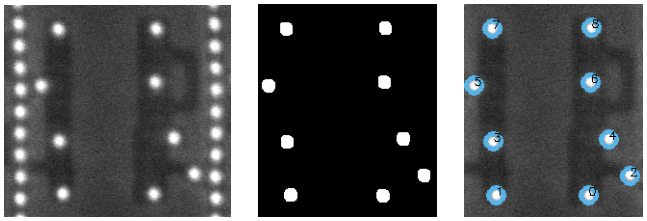
3.5. BLUE TEAM Decision Algorithms

The purpose of the decision algorithms is to distinguish either between filler cells and regular standard cells or between different functional standard cells. Processing each cell instance through any of the presented algorithms yields a number of candidate cells that are either true positives or false positives. A true positive means that a particular standard cell does not match the cell type it is labeled with. A false positive means that a significant difference is found from the *cell template*, but the cells are actually identical. A *cell template* is the first detected instance of a cell type that is not obstructed and where all important features are clearly visible.

To minimize the ratio of false positives to true positives, each of the algorithms is configured with a set of parameters. In general, parameter sets that lead to a relatively small number of positives are preferable to limit the manual evaluation effort. Of course, care must be taken to avoid discarding true positives in an attempt to reduce the number of candidates. Optimal parameters may be determined manually. However, suitable parameters can also be determined automatically by a trial-and-error procedure. The performance of automated parameter tuning can be further improved by providing already known true positives (e. g., from previous algorithm runs). As all tile images of the entire chip are recorded with the same imaging method and stem from the same chip, it is usually sufficient to optimize the parameters on a single tile image and use this optimized parameter set for all tiles. If there are quality differences in certain regions of the chip (e. g., on the border of the chip), the parameters must be verified for these regions. If the algorithms yield too many positives for manual review, this is an indicator that the parameters need to be further optimized to account for these quality differences.

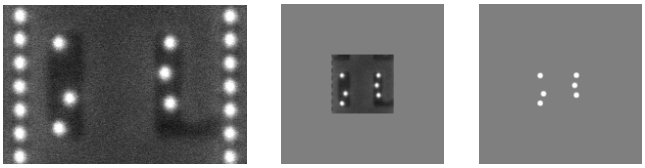
The decision algorithms directly or indirectly extract typical features that can distinguish between filler cells and standard cells, or between different functional cells. These distinguishing features include the number and position of vias, the tracks within standard cells, the orientation of power lines, and the occurrence of wells.

3.5.1. Via Detection Algorithm. In the first step, we only distinguish between a valid filler cell and modified filler cell, which in itself is a functional standard cell. To do this, we leverage the fact that each functional standard cell contains



(a) (b) (c)

Figure 3. Steps of the via detection algorithm. (a) shows the cell image, (b) the preprocessed and thresholded image of the cell, and (c) the detected vias as blue circles superimposed on the cell.



(a) (b) (c)

Figure 4. Template preparations of the template matching algorithm. (a) shows the cell image, (b) the preprocessed cell template image and (c) the via mask for template matching involving via masks.

at least two vias (1 input and 1 output). In contrast, filler cells do not contain vias.

Vias are of circular shape and typically have the same diameter within their respective fabrication technology. Also, tungsten vias appear much brighter than other structures on SEM images taken from the backside of the chip.

Any slight appearance of a via above a set threshold will be detected as possible candidate for a functional standard cell, i. e., modified filler cell.

A straightforward idea is to implement this in an algorithm that can determine the probability that bright spots are vias, as well as their position and size. Specifically, the algorithm finds bright spots through blurring, adaptive binary thresholding, eroding, dilation, and then performs an analysis of the connected components leading to the bounding boxes and the minimum perimeter of each contour. All of the above actions are already implemented in the OpenCV library [29], resulting in a rather lightweight Python implementation that can be found in the provided source code. An example cell analysis is showcased in Figure 3. Furthermore, the algorithm extracts the stochastic variance of the bounding rectangle of each detected via candidate to determine its overall contrast with other parts of the cell. If a via is clearly visible, the variance is above a certain value. Then, a radial gradient of the same size as the via is correlated with the detected via candidate to define the roundness of the via. Only good correlation values can confirm that the selected spot actually is circular and thus is a via.

A cell labeled as a filler cell is considered a candidate if it contains more than a certain number of vias. As filler cells usually do not have vias and regular standard cells have at least two vias connecting input and output, a threshold of one or two is viable. The resulting data can also be used to determine which cell type the via arrangement matches.

3.5.2. Template Matching Algorithm. Although the results of the previous algorithm are sufficient for evaluating the conformity of filler cells, they cannot be used directly for detecting standard cells replaced by an instance of a different standard cell type. Multiple instances of a standard cell always have the same characteristics (up to optional axis flips and right-angle rotations), because the labels are direct placeholders for the structure to be fabricated on the chip. Here, we assume that we can sufficiently distinguish the different standard cells based on the location of vias and other features in each image of the standard cell. If a cell deviates from the contained features of other appearances above a specified threshold, it is a candidate for a potential replacement manipulation. For this purpose, we utilize template matching between cell templates and each sample of the same cell type. If the distance between template and sample (i. e., the best correlation value of the template matching) is too large, the cell is considered a candidate.

Technically, the template matching is used in such a way that the cell template is expanded with a 50% gray border that does not affect correlation values. Due to inaccuracies in the mapping between design file and real images, the cell cutouts may be shifted. The gray border allows for better correlation when the best template match is positioned slightly outside the boundaries of the template image.

To further avoid shifted cutouts, a power line detection algorithm is used to fix the cell edges in one axis (for a detailed description see Section 3.4.2). The correction of the possible displacement of the axis perpendicular to power lines is implemented by iterative shifting of the cutout in a defined interval and step size. At each iteration, the template matching is performed until the distance between template and sample is sufficiently small. Also, both template and sample are slightly blurred before template matching with the normalized sum of squared difference algorithm [30] to improve the results. If after all iterations the distance is still too large, the cell is considered as a candidate.

In a variant of the template matching, another decision algorithm is executed if the cell is recognized as identical to the template. In this variant, a mask of vias of cell and template is generated using the via detection algorithm described in Section 3.5.1. The detected vias are represented by white radial gradients of a parameterized size on a 50% grey background. Now exactly the same template matching algorithm is executed with these via masks. With adjusted thresholds, candidates that previously went undetected can now be detected in certain cases because interference from other parts and the general noise are now reduced. Template images of an example run including the via mask are shown in Figure 4.

Once a candidate is found, it can be matched against the entire database of known cell templates using the same algorithm to find the best match. The result should be the suggested type of the replaced cell. If the cell type matches the label, i. e., there is no better match than the poor match to the original label, the candidate is likely a false positive.

4. Results

Here we present the findings of our real-world hardware Trojan detection case study. We first summarize the general results applicable to all four analyzed ICs before presenting the details for each case study individually. Lastly, we provide information on our datasets that can be leveraged to replicate our experiments or perform further experiments.

4.1. Overview and Summary

The quality of the results strongly depends on the accuracy of the detection algorithm, which is based on the choice of parameters and the image quality itself. It seems obvious that it is easier to find replaced filler cells than replaced functional standard cells. This is due to the additional challenge of correlating via positions and other visible features in the second case. When searching for replaced filler cells the sheer existence of vias is sufficient to count the cell as a candidate. However, if vias cannot be detected with high confidence, there is always the difficulty of choosing a good balance between more vias being incorrectly detected (e. g., caused by noise or small particles) and the possibility of overlooking single vias. Missing a feature such as a via is highly undesirable, as it might result in the worst case scenario of missing a true positive candidate. As the smallest replacements usually differ only in the position of a single via, it would be critical to detect all vias and their positions with sufficient accuracy. Thus, the only two options for improving the results in cases like these are either to take new images with better quality or to accept more false positive candidates.

Image Quality. As the image quality of the SEM images is of utmost importance for the following image processing steps we manually adjusted the contrast and electron beam acceleration to highlight underlying vias without too much blurry effects. An acceleration voltage between $10\,\mathrm{kV}$ and $15\,\mathrm{kV}$ was sufficient, while increasing the dwell time to obtain sharp images for the human eye. Often some areas are blurry as the wet etching process is not perfectly even. Due to the various manual steps required for each sample preparation, it is unlikely that an automated procedure can be used. The horizontal field width was set to make vias visible with a diameter of at least 20 to 30 pixels to achieve sufficient image quality for the decision algorithms.

During the acquisition of the SEM images of the small $28\,\mathrm{nm}$ node we encountered a drop in the image quality when decreasing the horizontal field view for a tile to $13.4\,\mu\mathrm{m}$. During a long scanning process, the image alignment and focus shifted, resulting in blurry, somewhat unaligned images. This result may be due to different factors, including an uneven sample preparation, charging effects of the sample, or the sum of inaccuracies in the stage movement. Physical properties like the vacuum, the selected detector, and beam currents become critical for small-size technologies. Although the imaging device software allows extrapolation by measuring multiple elevation points and adjusts the XY-axis sampling accordingly, these image quality problems could not be entirely avoided. Possible remedies could result in longer acquisition times and include dividing the chip into multiple regions, adding an additional sample preparation step to circumvent charging effects, using multiple extrapolation points, or using a different SEM.

Stitching Quality. Due to imaging effects, it appeared that the chips were bumped in one direction towards the center even with very good stitching. This was found to be particularly relevant in small structure sizes. However, with correction values between calculated cell coordinates and real images that were measured every fifth tile (in total up to 169 measurements per chip) correction values could be interpolated and determined for every tile which is sufficient for cutting out cells. Additional corrections could be realized using the power line detector described in Section 3.4.2.

Experimental Setting. In our experiments, the Red Team provided the Blue Team with a few physical samples and a design file for each chip in which four filler cell replacements and six standard cell replacements had been emulated (see Section 3.2). The Blue Team acquired SEM images of all of the four ASICs with structure sizes of $90\,\mathrm{nm}$, $65\,\mathrm{nm}$, $40\,\mathrm{nm}$, and $28\,\mathrm{nm}$ and used them to find the cell replacements in the design files (see sections 3.3, to 3.5). An important step at the beginning of the experiments was to find a good parameter set that would produce as few false-positive candidates as possible, but still detect as many modifications as possible. Variable parameters depend on the appearance of SEM images. Some of them can be directly extracted from the images (e. g., typical via size) while others need to be determined iteratively (e. g., correlation value thresholds). The iteratively determined parameters can be found in an automated way by observing the number of candidates on a few tiles. It seems feasible to start with a set of parameters that leads to very few candidates in relation to the total number of cells. When modifications are found while maintaining a relatively small number of false-positive candidates, the parameter set is considered good. If it is not known whether or how many modifications are present, artificial replacements such as those made for this work (e. g., replacing an XOR gate with an XNOR gate which is a subtle modification) can support finding good parameters. After each algorithm run, we classified all resulting candidates to identify why these were selected. Details of the ASIC analyzed by the Blue Team are presented in Table 1. Concrete results on detected cell candidates for modifications are reported in Table 2, which also contains the actual runtime of the respective experiments. The runtime was measured on a modern laptop computer with an Intel Core i7-8665U processor, and the image data was stored on an external SanDisk T7 SSD.

An example subset of true positive candidates for each chip and type of manipulation is shown in Figure 5. A complete overview of all true positives and false negatives is showcased in the Appendix. The individual types of false positives are discussed separately for each chip in the following sections.

Table 1. DETAILS OF THE ANALYZED ASICS AS DETERMINED BY THE BLUE TEAM.

	90 nm	65 nm	40 nm	28 nm
Number of Functional Standard Cells	222 963	222 430	446 498	583 901
Number of Filler Cells	230 887	348 630	471 321	883 950
Total Number of Cells in Standard Cell Area	453 850	571 060	917 819	1 467 851
Different Types of Functional Standard Cells	460	602	678	822
Different Types of Filler Cells	7	7	24	21
Total Area	$3.834\,\mathrm{mm}^2$	$3.771\,\mathrm{mm}^2$	$2.826\,\mathrm{mm}^2$	$1.901\,\mathrm{mm}^2$
Total Standard Cell Area	$2.089\,\mathrm{mm}^2$	$1.848\,\mathrm{mm}^2$	$1.052\,\mathrm{mm}^2$	$0.962\,\mathrm{mm}^2$
Average Cell Area	$4.602\,\mu\mathrm{m}^2$	$3.237\,\mu\mathrm{m}^2$	$1.146\,\mu\mathrm{m}^2$	$0.656\,\mu\mathrm{m}^2$
Pixel Size	12.207 nm	8.545 nm	4.883 nm	4.883 nm
Total Number of Tiles	$45 \times 46 = 2070$	$65 \times 65 = 4225$	$96 \times 95 = 9120$	$78 \times 78 = 6084$
Number of Tiles Covering Standard Cell Area	$33 \times 33 = 1089$	$44 \times 44 = 1936$	$58 \times 58 = 3364$	$56 \times 56 = 3136$
Acquisition Duration	33:44 h	22:38 h	53:05 h	35:52 h

Table 2. DETECTION RESULTS FOR THE FILLER CELL REPLACEMENTS (TOP) AND THE STANDARD CELL REPLACEMENTS (BOTTOM) ON ALL CHIPS.

	90 nm	65 nm	40 nm	28 nm
Total True Positives	4	4	4	4
False Negatives	0	0	0	0
False Positives (Debris / Dust)	0 (0 %)	0 (0 %)	4 (50 %)	10 (47.62 %)
False Positives (Low / High Contrast)	0 (0 %)	0 (0 %)	0 (0 %)	7 (33.33 %)
Total False Positives	0	0	4	17
Runtime	0:15 h	1:50 h	1:17 h	0:55 h

	90 nm	65 nm	40 nm	28 nm
Total True Positives	6	6	6	3
False Negatives	0	0	0	**3**
False Positives (Debris / Dust)	2 (1.41 %)	6 (50 %)	11 (64.71 %)	128 (36.99 %)
False Positives (Low / High Contrast)	98 (69.01 %)	0 (0 %)	0 (0 %)	7 (2.02 %)
False Positives (Blur)	36 (25.35 %)	0 (0 %)	0 (0 %)	51 (14.74 %)
False Positives (Cell Stitching Error)	0 (0 %)	0 (0 %)	0 (0 %)	70 (20.23 %)
False Positives (Power Line Stitching Error)	0 (0 %)	0 (0 %)	0 (0 %)	87 (25.14 %)
Total False Positives	136	6	11	343
Runtime	1:30 h	1:32 h	4:09 h	2:41 h

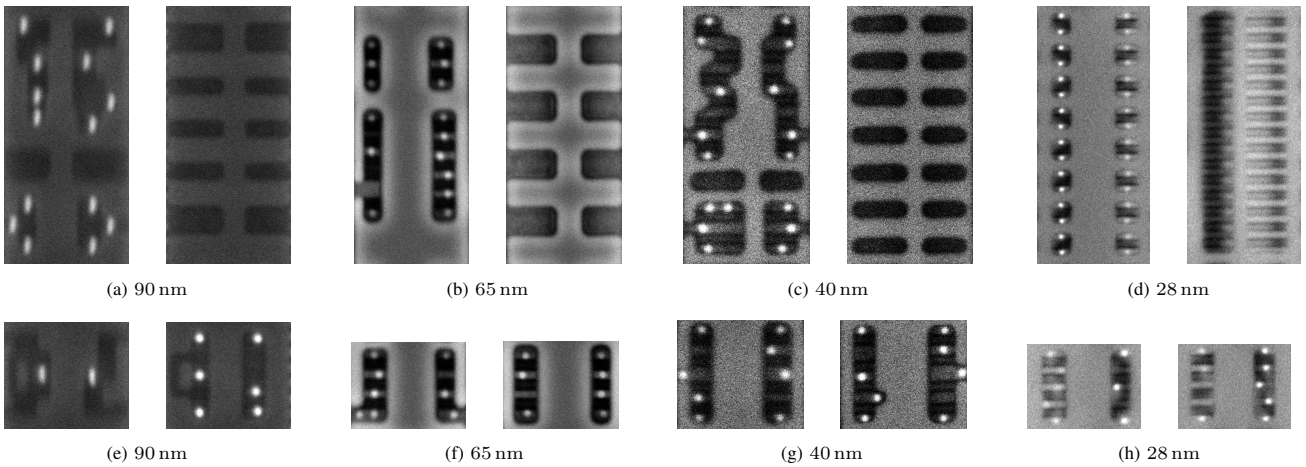
4.2. The 90 nm Chip

Filler Cell Replacements. All four replacements could be found by applying the via detection algorithms on all filler cells. As the vias could be detected very reliably on the 90 nm chip, we trimmed the edges of the cell bounding boxes. This reduces the probability of induced false positive results when cell boundaries are not perfectly laid over the images. On all replaced cells at least one of the vias could be found, which is sufficient as typically at least one via is positioned close to the middle of every cell type that is no filler cell. No false positive results are obtained in this algorithm run.

Functional Standard Cell Replacements. All replacements could be found by applying the template matching algorithm on all standard cells, excluding the filler cells which were found during via detection. All but one (described later) of the true positives are clear no-matches and thus could be detected without any issues. Within template matching it was not required to crop the cell images as eventual inaccuracies in the overlay of bounding boxes wouldn't lead to any false positives here.

By intuition the total number of false positive candidates would be larger the smaller the technology gets or the worse the image quality would be. However, also subtle cell replacements lead to the requirement of more sensitive parameters to find these replacements. On the 90 nm chip it incidentally occurred (due to the random choice of cells to replace) that a very subtle modifications was caused, i. e., a cell was replaced by another cell with similar appearance. This one interesting case could not be detected with the normal template matching algorithm but was indeed found using the via mask correlation. The replacement is depicted in Figure 6.

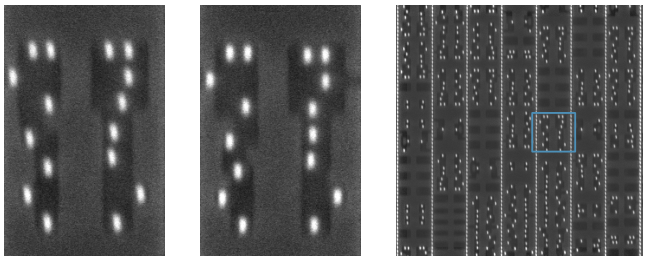
Manually examining the total number of 142 candidates including the six true positives led to the following classification. Two single cell candidates were covered by dust or debris caused by not imaging in a cleanroom environment. 98 other cells in one edge of the chip have been affected by processing steps and appeared too dark as seen in Figure 7a. Figure 7b shows the same cell type but on another position



(a) 90 nm (b) 65 nm (c) 40 nm (d) 28 nm

(e) 90 nm (f) 65 nm (g) 40 nm (h) 28 nm

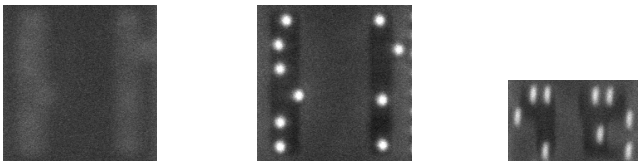
Figure 5. Top: Exemplary true positives of the filler cell replacement detection. Bottom: Exemplary true positives of the standard cell replacement detection. Both: The actual cell (left) and an expected instance of the same cell type (right) are shown.



(a) (b) (c)

Figure 6. (a) and (b) are very similar looking cells. In (c) there is an excerpt of the chip image. The blue border is the same as (a) but labeled as (b).

of the chip. Features there are more clearly visible. Likely caused by imperfect preparation or SEM settings, some tile images are blurry, also inducing problems with the algorithms. One example of the total 36 false positives caused by blur is given in Figure 7c.



(a) (b) (c)

Figure 7. Examples of false positives caused in the imaging process on the 90 nm chip. (a) shows bad contrast vs. its template (b). (c) shows a blur.

After all, checking the 142 candidates costs a human expert only 6-7 minutes in our experience. False positives caused by low-quality images are usually easy to identify for the human eye.

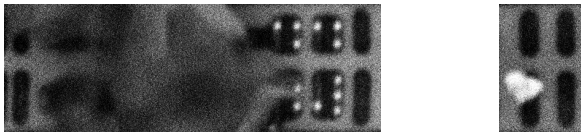
4.3. The 65 nm Chip

Filler Cell Replacements. All four replacements could be found by applying the via detection algorithms on all filler cells. On all of the replaced cells, multiple vias could be found. No false positive results have been obtained in this algorithm run.

Functional Standard Cell Replacements. All replacements could be found by applying the template matching algorithm on all standard cells, excluding the filler cells which were found during via detection. Manually examining the total number of 12 candidates including the six true positives led to the following classification. All six false positive candidates are caused by dust or debris that covers parts of the cell images. The disturbances lead to mismatch results during template matching, even though parts of the original cell image are clearly visible in the cell cutout. In all but one of the false positive cases a cell identification algorithm detects the correct cell template as best match. The cell identification algorithm runs template matching against all cell templates of the same dimensions and returns the cell type that has the highest confidence to be the correct fit. Since in one case an incorrect cell type was identified as the best match, this method is not perfectly reliable when trying to recognize false positive candidates in an automated fashion. In our experience, a human expert usually performs this task with better accuracy.

4.4. The 40 nm Chip

Filler Cell Replacements. All four replacements could be found by applying the via detection algorithms on all filler cells. On all of the replaced cells, multiple vias could be found. The total four false positive candidates are caused by dust or debris that covers parts of the filler cells, as one example shows in Figure 8b.



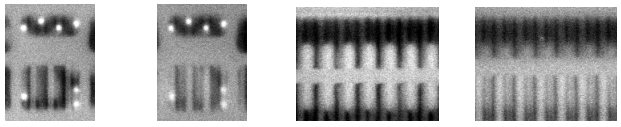
(a) (b)

Figure 8. Two examples of false positives caused by debris on the 40 nm chip. (a) shows debris over a standard cell, while (b) shows debris over a filler cell.

Functional Standard Cell Replacements. All replacements could be found by applying the template matching algorithm on via masks of all standard cells, excluding the filler cells which were found during via detection. Further, the regular template matching could be used to first sort out false positive candidates that would have been detected as candidates using only the via mask algorithm. Manually examining the total number of 17 candidates including the six true positives led to the following classification. All 11 false positive candidates are caused by dust or debris that covers parts of the cell images, as one example shows in Figure 8a. In five of the false positive candidates the debris was so large that it covered the cell cutout completely, leaving it unclear if behind the disturbance there is the original cell or another cell modification that would have been left undetected. In a real world scenario, this would mean that another image acquisition run of the disturbed area is required. In this study we disregard another run for the sake of cost and time required. The partly disturbed six candidates can be manually identified as the labeled cell types, whilst four could also be identified automatically using the cell identification algorithm.

4.5. The 28 nm Chip

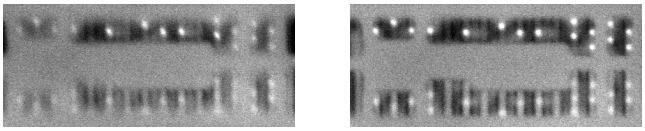
Filler Cell Replacements. All four replacements could be found by applying the via detection algorithms on all filler cells. On all of the replaced cells, at least one of the vias could be found. Manually examining the total number of 21 candidates including the four true positives led to the following classification. 10 of the false positive candidates are caused by dust or debris that covers parts of the filler cells. Seven of the false positive candidates are caused by a too high contrast, as one example shows in Figure 9c. Even though the contrast is slightly better in those regions, it makes the detection obviously more difficult, as the parameters of the via detection algorithm have to be altered in such areas. With a general per-chip parameter set, these false positive candidates inevitably occur.



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

Figure 9. Two examples of false positives caused by contrast on the 28 nm chip. (a) shows bad contrast of a standard cell vs. its template (b). (c) shows bad contrast of a filler cell vs. its template (d).

Functional Standard Cell Replacements. Only three out of six replacements could be found by applying the template matching algorithm on via masks of all standard cells, excluding the filler cells which were found during via detection. The regular template matching was not required hereby, as via mask correlation was meaningful enough and did not lead to more false positive results than with a regular template matching as a pre-filter, similar to the 40 nm chip analysis. Manually examining the total number of 346 candidates including the three true positives led to the following classification. 53.75 % of the candidates are caused by small variations in the imaging step. 128 cell candidates were covered by dust or debris. Some cells are obstructed completely while others can still be identified as unmodified. Seven of the false positive candidates are caused by a too high contrast, as one example shows in Figure 9a. 51 of the false positive candidates appear blurry as some tile images appear to be recorded with a blur. One example of a false positive caused by blur is given in Figure 10a. Because of the generally lower image quality, the power line detector described in Section 3.4.2 could not be applied successfully. Even with optimized parameters, the power lines were often detected too inaccurate or completely wrong (e. g., moved into the center of cell rows), thus the correction of cell cutouts by detected power lines was omitted completely. However, as cutout corrections and stitching quality in general was not very accurate, numerous false positive candidates are caused by wrong cell cutouts. 70 false positives reveal parts of neighboring cells in the axis of power lines, while further 87 false positives show cell cutouts drifted over adjacent power line rows. Adjusting our algorithms to better deal with lower-quality images is certainly among our future works considered. Given the fact that IC manufacturing is still constantly moving towards smaller dimensions it becomes more and more difficult to scan the tiny structures in high quality.



(a) (b)

Figure 10. An example of a false positive caused by blur on the 28 nm chip (a) vs. its template (b).

Given that the Blue Team was unable to detect three of the six cell replacements, we retrospectively worked with the Red Team to determine the correlation threshold necessary to find these three missed candidates (see Appendix A.4). Assuming the correlation value of the most subtle manipulation, an analyst would have had to manually check about 78 800 candidates, which does not seem feasible.

4.6. Publication of the Experimental Datasets

To ensure reproducibility of our experiments and to allow interested researchers to conduct their own experiments, we have made all source code, SEM images, and abstracted layout data available online. It can be found at https://github.com/emsec/ChipSuite. The GDSII layout files contain only rectangles representing the bounding boxes of the original cell instances and cell type labels. In this way, we protect any Intellectual Property (IP) that might otherwise be contained in the layout files, while still providing all the crucial information required for our experiments. The code is designed to run with Python 3.10 and requires only a few additional standard modules (opencv, imutils, numpy, and gdspy). The images are the original SEM backside images in TIFF format with a total file size of 300.2 GiB.

5. Related Work

In the following section, we provide an overview of existing hardware Trojan taxonomies before presenting examples of Trojans that can be inserted within the malicious foundry model. Lastly, we review methods for detecting Trojans, with a particular focus on previous work on reverse-engineering based detection.

5.1. Hardware Trojan Overview and Classification

A hardware Trojan can be characterized as a malicious modification of an IC which has been intentionally inserted [31]. In 2008, Adee postulated the threat of hardware Trojans that act as *kill switches* [5]. Three years earlier, in 2005, DARPA launched its "TRUST in Integrated Circuits" research program to study the trustworthiness of military-grade ICs built under untrusted conditions [32]. Back then, they explicitly highlighted the risk of ICs with additional (malicious) circuitry and identified the foundry as the most important untrusted party in the globalized IC supply chain [33]. One of the biggest challenges DARPA named for building trust in ICs was performing destructive reverse engineering in a cost- and time-efficient manner.

Since 2008, several taxonomies for the classification of hardware Trojans have been proposed and continuously refined. In the most basic model, hardware Trojans consist of an activation mechanism – called trigger logic – and a payload logic [34], [35]. Possible triggers include always-on, time-based, or user input-based triggers, while examples of payloads range from denial of service over information leakage to performance degradation. Other taxonomies classify hardware Trojans according to their physical, activation (trigger) and action (payload) characteristics [36], [31]. Karri et al. proposed an even broader taxonomy in which they systematically classified hardware Trojans according to their insertion phase, abstraction level, activation mechanism (trigger), effects (payload), and on-chip location [7]. Trojan insertion phases include specification, design, as well as fabrication or assembly, and examples for abstraction levels range from system level over gate level down to the physical level. As introduced in Section 1 and discussed in Section 2, our work is concerned with the realistic scenario of small hardware Trojans consisting of added or modified gates (abstraction level) inserted during fabrication. In our work, we did not place any restrictions on the particular design of the trigger, the payload, or the Trojan's location on the chip.

5.2. Hardware Trojan Design and Implementations

Here, we summarize previous work showing the practical feasibility of hardware Trojans inserted by malicious foundries. In 2013, Muehlberghuber et al. described a Blue Team vs. Red Team hardware Trojan analysis, where the Red Team acted as a malicious foundry and inserted a Trojan circuit to a $180\,\mathrm{nm}$ IC adding 0.5% overhead [6]. The Blue Team applied IC fingerprinting techniques and performed side-channel analyses to successfully distinguish trojanized from Trojan-free ICs. In 2016, Yang et al. proposed *A2*, a Trojan forcing a flip-flop to a desired value [37]. Their Trojan can be implemented with an overhead of only a single gate and can be placed by a malicious foundry into the open spaces of an already placed and routed design. They fabricated an IC containing their Trojan in an open-source processor using $65\,\mathrm{nm}$ CMOS technology and claimed that a defender delayering the chip and imaging it with a SEM would not be able to distinguish the added malicious gate from the other gates in the design. In 2020, Ghandali et al. presented a hardware Trojan consisting of delay gates that introduce path delay faults to a side-channel protected block cipher, which was implemented in two different $65\,\mathrm{nm}$ and $90\,\mathrm{nm}$ ASICs [38]. Subsequently, they showed that triggering the Trojan makes the ASIC prototypes vulnerable to side-channel attacks. In 2021, Perez et al. elaborated how a malicious foundry can insert a side-channel hardware Trojan after physical synthesis using the Engineering Change Order (ECO) feature of commercial Electronic Design Automation (EDA) tools to modify or insert additional logic in a finalized layout [9]. Subsequently, Perez and Pagliarini practically demonstrated Trojan insertion via ECO by injecting four different side-channel Trojans that leak cryptographic keys into a commercial $65\,\mathrm{nm}$ chip [10], [11]. In 2022, Almeida et al. introduced a ransomware Trojan that can be implemented by a malicious foundry in a System-on-a-Chip (SoC), increasing the area utilization factor from 59.97% to 60.7% [12].

5.3. Hardware Trojan Detection

Several methods have been proposed to detect hardware Trojans. The most prominent non-destructive detection techniques are logic testing and side-channel analysis [31], [35]. Logic testing involves applying carefully crafted digital stimuli to circuit inputs and inspecting the outputs for unexpected behavior [39]. Trojan detection methods via side-channel analysis assume that insertion of a hardware Trojan affects the power consumed by the circuit or influences the delay of certain circuit paths [40]. However, it is unlikely that stealthy hardware Trojans can be detected in a non-destructive manner [3]. Thus, methods based on visual inspection are an important tool of choice for Trojan detection.

Below, we summarize the works that have already leveraged reverse engineering and visual inspection to detect hardware Trojans in ICs: In 2014, Courbon et al. analyzed a smart card IC consisting of 15 000 standard cells, manufactured in $130\,\mathrm{nm}$ technology covering an area of $0.49\,\mathrm{mm}^2$ [13]. After front side preparation, they acquired 64 SEM images of the standard cell layer and were able to detect all four standard cells that had been "manually added to the bottom left area" of the chip when compared to the original design files. One year later, Courbon et al. conducted another experiment with an IC composed of 40 000 standard cells fabricated in $180\,\mathrm{nm}$ technology covering an area of $1\,\mathrm{mm}^2$ [14]. They successfully discovered a hardware Trojan that added about 190 gate equivalents to the circuit when they compared SEM images of a trojanized IC against those of a non-trojanized IC. In 2018, Vashistha et al. analyzed a smart card die in a non-disclosed technology size and standard cell count with an

area of $2.25\,\mathrm{mm}^2$ for hardware Trojans [41], [42]. They manually added their Trojan cells to the SEM images captured from the backside and achieved up to 98% detection accuracy for nine different types of standard cells.

Another branch of research is concerned with the verification of the physical layout – primarily in the context routing on the metal layers – of ICs. Singla et al. presented several algorithms to pre-process and compare original design and reverse-engineered physical layout data [43]. On this basis, Lippmann et al. overlayed 130 images acquired from one metal layer of a $40\,\mathrm{nm}$ chip with design data and achieved high accuracy, leaving about one percent of the data for manual inspection [44]. Their method also allowed them to detect a malicious manipulation artificially inserted into a tile of their original design. In 2021, Ludwig et al. introduced the *ViTaL* framework for physical layout verification building upon the works of Singla, Lippmann et al. [45]. They evaluated ViTaL on three metal layers of a $40\,\mathrm{nm}$ IC with an area of $0.036\,\mathrm{mm}^2$ with accuracies between 95% and 99%. We point out here that novel, non-invasive scanning methods based on X-Rays [17] may lead to further progress in routing verification.

None of the above works implemented a strict Red Team vs. Blue Team approach, which could have introduced bias into their results. Such Red Team vs. Blue Team approaches were first proposed and conducted by Rajendran et al. [46] and Muehlberghuber et al. [6] for other – non-invasive – Trojan detection techniques.

6. Conclusion

In this work we have presented a comprehensive hardware Trojan detection case study based on four different digital ICs manufactured in progressively smaller CMOS process technologies ($90\,\mathrm{nm}$, $65\,\mathrm{nm}$, $40\,\mathrm{nm}$, and $28\,\mathrm{nm}$). Our detection is based on GDSII–vs–SEM-image comparisons using simple and scalable image processing techniques and has been conducted using a Red Team vs. Blue Team strategy to ensure an unbiased assessment. Our analysis indicates that those types of Trojans which are easier to integrate (e. g., through automated insertion via ECO flow) are also less difficult to detect. This remains true regardless of how small their overhead is, as every single cell modification could be detected. Across all four devices, our algorithms isolated 37 out of a total of 1.9 million filler cells as potential candidates for such a Trojan insertion. All 16 true positives were included in this short list of candidates. The second type of hardware Trojan, which we have focused on in this work, requires a more skilled and knowledgeable adversary, but also proved to be more subtle and unobtrusive. The insertion strategy is based on replacing existing functional standard cells by other functional standard cells to realize the Trojan. On all four ICs combined, our algorithms isolated 517 out of a total of 1.5 million functional standard cells as potential candidates for such a Trojan insertion. 21 of 24 true positives were included in this list of candidates. According to our experience, a list of 517 candidates can be manually checked to separate true and false positives in less than half an hour by a qualified individual (unless a cell is fully obstructed by debris – then no classification is possible). The duration for image acquisition falls in the range of one to three days per device, the detection algorithms provide runtimes between minutes and a few hours. Virtually all false positive and negative results can be attributed to insufficient image quality in some parts of the ICs. The majority of them occurs for the smaller nanometer technologies, as it becomes more and more difficult to capture high-resolution images of those shrinking structures due to focus and alignment shifts during the scans. Additionally we have noticed that the complexity of the standard cell library, which increases in advanced technologies, also plays a role in making the detection more difficult. Finally, particles of dust and debris obstruct larger fragments of cells in smaller technologies, simply because of their decreased size. We make our datasets and our source code available to the public for independent verification of our results and to spark further investigations in this important field of research.

6.1. Future Work

We believe that our detection can likely be improved by using image recognition software based on deep learning, of course at the cost of a significantly longer run time. As we provide the underlying datasets, future studies can easily compare their results to ours. The imaging itself could be improved by performing the SEM scans in a cleanroom and/or using a more advanced scanning device. Our setup may reach its limits when moving significantly below the $28\,\mathrm{nm}$ node. Yet we believe that high-precision imaging devices, such as those used to control and operate today's semiconductor lithography systems, should be able to scan even the most advanced ICs with sufficient quality for an analysis similar to ours. The cost, effort and required expertise for such an investigation, however, would be inversely proportional to the technology's feature size.

Acknowledgements

We thank Sebastian Sester-Wehle, a member of the BKA's KT52 team, for milling and the KT44 and KT41 staff for assistance with chemical sample preparation and chip imaging. We are grateful to the Netherlands Forensic Institute hardware laboratory for milling additional samples during a machine breakdown on our side. This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA — 390781972, by the European Union (EU) through the ERC project SWORD (724725), and by the EU and the Walloon Region through the FEDER project USERMedia (501907-379156).

References

[1] Y.-C. Chou, C.-T. Cheng, F.-C. Yang, and Y.-Y. Liang, "Evaluating alternative capacity strategies in semiconductor manufacturing under uncertain demand and price scenarios," *International journal of production economics*, vol. 105, no. 2, pp. 591–606, 2007. [Online]. Available: https://doi.org/10.1016/j.ijpe.2006.05.006

[2] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Trans. Design Autom. Electr. Syst.*, vol. 22, no. 1, pp. 6:1–6:23, 2016. [Online]. Available: https://doi.org/10.1145/2906147

[3] Defense Science Board Washington DC, "Defense science board task force on high performance microchip supply," US DoD, Tech. Rep., 2 2005. [Online]. Available: https://dsb.cto.mil/reports/2000s/ADA435563.pdf

[4] J. F. Miller, "Supply chain attack framework and attack patterns," The MITRE Corporation, Tech. Rep., 12 2013. [Online]. Available: https://www.mitre.org/sites/default/files/publications/supply-chain-attack-framework-14-0228.pdf

[5] S. Adee, "The hunt for the kill switch," *IEEE SpEctrum*, vol. 45, no. 5, pp. 34–39, 2008. [Online]. Available: https://spectrum.ieee.org/the-hunt-for-the-kill-switch

[6] M. Muehlberghuber, F. K. Gürkaynak, T. Korak, P. Dunst, and M. Hutter, "Red team vs. blue team hardware trojan analysis: detection of a hardware trojan on an actual ASIC," in *HASP 2013, The Second Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, June 23-24, 2013*, R. B. Lee and W. Shi, Eds. ACM, 2013, p. 1. [Online]. Available: https://doi.org/10.1145/2487726.2487727

[7] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, 2010. [Online]. Available: https://doi.org/10.1109/MC.2010.299

[8] K. Xiao and M. Tehranipoor, "BISA: built-in self-authentication for preventing hardware trojan insertion," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013, Austin, TX, USA, June 2-3, 2013*. IEEE Computer Society, 2013, pp. 45–50. [Online]. Available: https://doi.org/10.1109/HST.2013.6581564

[9] T. D. Perez, M. Imran, P. Vaz, and S. Pagliarini, "Side-channel trojan insertion - a practical foundry-side attack via ECO," in *IEEE International Symposium on Circuits and Systems, ISCAS 2021, Daegu, South Korea, May 22-28, 2021*. IEEE, 2021, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ISCAS51556.2021.9401481

[10] T. D. Perez and S. Pagliarini, "Hardware trojan insertion in finalized layouts: From methodology to a silicon demonstration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[11] ——, "A side-channel hardware trojan in 65nm cmos with 2μW precision and multi-bit leakage capability," in *27th Asia and South Pacific Design Automation Conference, ASP-DAC 2022, Taipei, Taiwan, January 17-20, 2022*. IEEE, 2022, pp. 9–10. [Online]. Available: https://doi.org/10.1109/ASP-DAC52403.2022.9712490

[12] F. Almeida, M. Imran, J. Raik, and S. Pagliarini, "Ransomware attack as hardware trojan: a feasibility and demonstration study," *IEEE Access*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9760411

[13] F. Courbon, P. Loubet-Moundi, J. J. A. Fournier, and A. Tria, "A high efficiency hardware trojan detection technique based on fast SEM imaging," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, W. Nebel and D. Atienza, Eds. ACM, 2015, pp. 788–793. [Online]. Available: http://dl.acm.org/citation.cfm?id=2755932

[14] ——, "SEMBA: A SEM based acquisition technique for fast invasive hardware trojan detection," in *European Conference on Circuit Theory and Design, ECCTD 2015, Trondheim, Norway, August 24-26, 2015*. IEEE, 2015, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ECCTD.2015.7300097

[15] M. Xue, C. Gu, W. Liu, S. Yu, and M. O'Neill, "Ten years of hardware trojans: a survey from the attacker's perspective," *IET Comput. Digit. Tech.*, vol. 14, no. 6, pp. 231–246, 2020. [Online]. Available: https://doi.org/10.1049/iet-cdt.2020.0041

[16] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, ser. Lecture Notes in Computer Science, C. Clavier and K. Gaj, Eds., vol. 5747. Springer, 2009, pp. 363–381. [Online]. Available: https://doi.org/10.1007/978-3-642-04138-9_26

[17] M. Holler, M. Odstrcil, M. Guizar-Sicairos, M. Lebugle, E. Müller, S. Finizio, G. Tinti, C. David, J. Zusman, W. Unglaub *et al.*, "Three-dimensional imaging of integrated circuits with macro- to nanoscale zoom," *Nature Electronics*, vol. 2, no. 10, pp. 464–470, 2019. [Online]. Available: https://www.nature.com/articles/s41928-019-0309-z

[18] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans," in *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, ser. Lecture Notes in Computer Science, G. Bertoni and J. Coron, Eds., vol. 8086. Springer, 2013, pp. 197–214. [Online]. Available: https://doi.org/10.1007/978-3-642-40349-1_12

[19] T. Sugawara, D. Suzuki, R. Fujii, S. Tawa, R. Hori, M. Shiozaki, and T. Fujino, "Reversing stealthy dopant-level circuits," in *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, L. Batina and M. Robshaw, Eds., vol. 8731. Springer, 2014, pp. 112–126. [Online]. Available: https://doi.org/10.1007/978-3-662-44709-3_7

[20] S. Ghandali, G. T. Becker, D. E. Holcomb, and C. Paar, "A design methodology for stealthy parametric trojans and its application to bug attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, ser. Lecture Notes in Computer Science, B. Gierlichs and A. Y. Poschmann, Eds., vol. 9813. Springer, 2016, pp. 625–647. [Online]. Available: https://doi.org/10.1007/978-3-662-53140-2_30

[21] T. Moos, "Physical security for next generation CMOS ICs," Ph.D. dissertation, Ruhr University Bochum, Germany, 2022. [Online]. Available: https://hss-opus.ub.ruhr-uni-bochum.de/opus4/frontdoor/index/index/docId/9275

[22] V. Korchnoy, "Investigation of choline hydroxide for selective silicon etch from a gate oxide failure analysis standpoint," *Conference Proceedings from the International Symposium for Testing and Failure Analysis*, 1 2002.

[23] J. Chalfoun, M. Majurski, T. Blattner, K. Bhadriraju, W. Keyrouz, P. Bajcsy, and M. Brady, "MIST: Accurate and scalable microscopy image stitching tool with stage modeling and error minimization," *Scientific Reports*, vol. 7, no. 1, p. 4988, 2017. [Online]. Available: https://doi.org/10.1038/s41598-017-04567-y

[24] D. Hörl, F. R. Rusak, F. Preusser, P. Tillberg, N. Randel, R. K. Chhetri, A. Cardona, P. J. Keller, H. Harz, H. Leonhardt, M. Treier, and S. Preibisch, "Bigstitcher: reconstructing high-resolution image datasets of cleared and expanded samples," *Nature Methods*, vol. 16, no. 9, p. 870–874, 2019. [Online]. Available: https://doi.org/10.1038/s41592-019-0501-0

[25] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, p. 676–682, 2012. [Online]. Available: https://doi.org/10.1038/nmeth.2019

[26] C. T. Rueden, J. E. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri, "Imagej2: Imagej for the next generation of scientific image data," *BMC Bioinform.*, vol. 18, no. 1, pp. 529:1–529:26, 2017. [Online]. Available: https://doi.org/10.1186/s12859-017-1934-z

[27] I. Sobel, "An isotropic 3x3 image gradient operator," *Presentation at Stanford A.I. Project*, 1968.

[28] P. V. C. Hough, *Method and means for recognizing complex patterns*. U.S. Patent 3,069,654, 1962.

[29] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[30] M. Hisham, S. N. Yaakob, R. Raof, A. A. Nazren, and N. Wafi, "Template matching using sum of squared difference and normalized cross correlation," in *2015 IEEE Student Conference on Research and Development (SCOReD)*, 2015, pp. 100–104. [Online]. Available: https://doi.org/10.1109/SCORED.2015.7449303

[31] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10–25, 2010. [Online]. Available: https://doi.org/10.1109/MDT.2010.7

[32] D. R. Collins, "TRUST, a proposed plan for trusted integrated circuits," Defense Advanced Research Projects Agency (DARPA), Tech. Rep., 2006. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA456459.pdf

[33] D. Collins, "DARPA "TRUST in IC's" effort (briefing charts)," Defense Advanced Research Projects Agency (DARPA), Tech. Rep., 2007. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA503809.pdf

[34] F. G. Wolff, C. A. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards trojan-free trusted ICs: Problem analysis and detection scheme," in *Design, Automation and Test in Europe, DATE 2008, Munich, Germany, March 10-14, 2008*, D. Sciuto, Ed. ACM, 2008, pp. 1362–1365. [Online]. Available: https://doi.org/10.1109/DATE.2008.4484928

[35] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014. [Online]. Available: https://doi.org/10.1109/JPROC.2014.2334493

[36] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2008, Anaheim, CA, USA, June 9, 2008. Proceedings*, M. Tehranipoor and J. Plusquellic, Eds. IEEE Computer Society, 2008, pp. 15–19. [Online]. Available: https://doi.org/10.1109/HST.2008.4559039

[37] K. Yang, M. Hicks, Q. Dong, T. M. Austin, and D. Sylvester, "A2: analog malicious hardware," in *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 2016, pp. 18–37. [Online]. Available: https://doi.org/10.1109/SP.2016.10

[38] S. Ghandali, T. Moos, A. Moradi, and C. Paar, "Side-channel hardware trojan for provably-secure sca-protected implementations," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 28, no. 6, pp. 1435–1448, 2020. [Online]. Available: https://doi.org/10.1109/TVLSI.2020.2982473

[39] R. S. Chakraborty, F. G. Wolff, S. Paul, C. A. Papachristou, and S. Bhunia, "MERO: A statistical approach for hardware trojan detection," in *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, ser. Lecture Notes in Computer Science, C. Clavier and K. Gaj, Eds., vol. 5747. Springer, 2009, pp. 396–410. [Online]. Available: https://doi.org/10.1007/978-3-642-04138-9_28

[40] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. G. Wolff, C. A. Papachristou, K. Roy, and S. Bhunia, "Hardware trojan detection by multiple-parameter side-channel analysis," *IEEE Trans. Computers*, vol. 62, no. 11, pp. 2183–2195, 2013. [Online]. Available: https://doi.org/10.1109/TC.2012.200

[41] N. Vashistha, M. T. Rahman, H. Shen, D. L. Woodard, N. Asadizanjani, and M. M. Tehranipoor, "Detecting hardware trojans inserted by untrusted foundry using physical inspection and advanced image processing," *J. Hardw. Syst. Secur.*, vol. 2, no. 4, pp. 333–344, 2018. [Online]. Available: https://doi.org/10.1007/s41635-018-0055-0

[42] N. Vashistha, H. Lu, Q. Shi, M. T. Rahman, H. Shen, D. L. Woodard, N. Asadizanjani, and M. Tehranipoor, "Trojan scanner: Detecting hardware trojans with rapid sem imaging combined with image processing and machine learning," in *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*. ASM International, 2018, p. 256. [Online]. Available: https://doi.org/10.31399/asm.cp.istfa2018p0256

[43] A. Singla, B. Lippmann, and H. Graeb, "Verification of physical chip layouts using GDSII design data," in *4th IEEE International Verification and Security Workshop, IVSW 2019, Rhodes Island, Greece, July 1-3, 2019*. IEEE, 2019, pp. 55–60. [Online]. Available: https://doi.org/10.1109/IVSW.2019.8854432

[44] B. Lippmann, N. Unverricht, A. Singla, M. Ludwig, M. Werner, P. Egger, A. Dübotzky, H. Gräb, H. A. Gieser, M. Rasche, and O. Kellermann, "Verification of physical designs using an integrated reverse engineering flow for nanoscale technologies," *Integr.*, vol. 71, pp. 11–29, 2020. [Online]. Available: https://doi.org/10.1016/j.vlsi.2019.11.005

[45] M. Ludwig, A.-C. Bette, and B. Lippmann, "Vital: Verifying trojan-free physical layouts through hardware reverse engineering," in *2021 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, 2021, pp. 1–8.

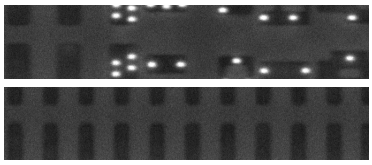
[46] J. Rajendran, V. Jyothi, and R. Karri, "Blue team red team approach to hardware trust assessment," in *IEEE 29th International Conference on Computer Design, ICCD 2011, Amherst, MA, USA, October 9-12, 2011*. IEEE Computer Society, 2011, pp. 285–288. [Online]. Available: https://doi.org/10.1109/ICCD.2011.6081410

Appendix A.
Full Disclosure of True Positive and False Negative Results

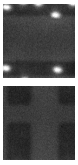
All 37 true positives and three false negatives are showcased below. The upper images each show the actual cell cutouts from the chip images, while the lower images each depict the cell templates according to the label from the respective design file. The difference score is the direct output value of the normalized sum of squared difference algorithm.

A.1. All Modifications of the 90 nm Chip

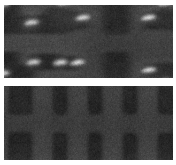
Detected True Positives: Filler Cell Replacements.



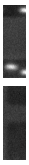
Detected Vias: 16



Detected Vias: 3

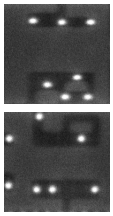


Detected Vias: 7

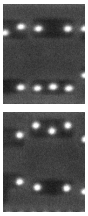


Detected Vias: 1

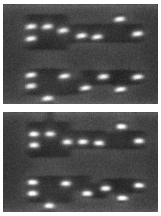
Detected True Positives: Functional Standard Cell Replacements.



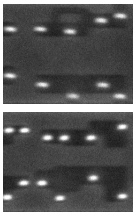
Difference Score: 0.213
Without via mask



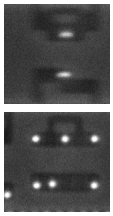
Difference Score: 0.197
Without via mask



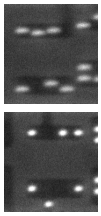
Difference Score: 0.008
With via mask



Difference Score: 0.227
Without via mask



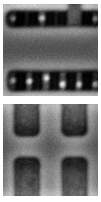
Difference Score: 0.168
Without via mask



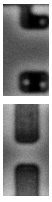
Difference Score: 0.186
Without via mask

A.2. All Modifications of the 65 nm Chip

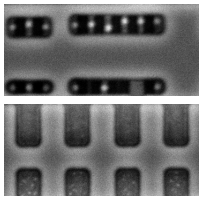
Detected True Positives: Filler Cell Replacements.



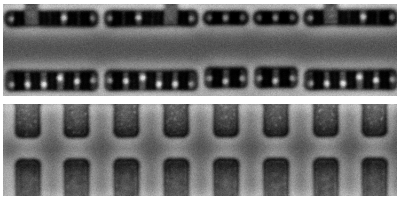
Detected Vias: 9



Detected Vias: 3

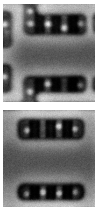


Detected Vias: 11

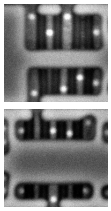


Detected Vias: 34

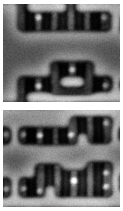
Detected True Positives: Functional Standard Cell Replacements.



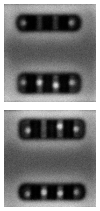
Difference Score: 0.1
Without via mask



Difference Score: 0.221
Without via mask

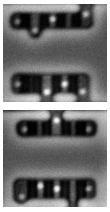


Difference Score: 0.268
Without via mask



Difference Score: 0.082
Without via mask

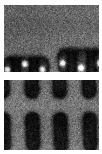
Difference Score: 0.257
Without via mask



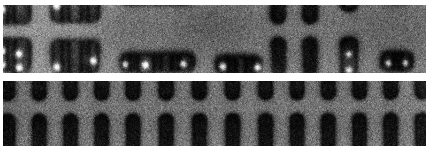
Difference Score: 0.152
Without via mask

A.3. All Modifications of the 40 nm Chip

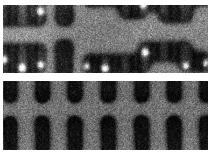
Detected True Positives: Filler Cell Replacements.



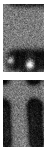
Detected Vias: 4



Detected Vias: 12

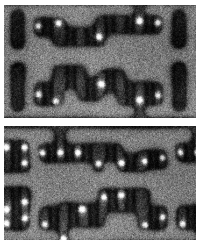


Detected Vias: 7

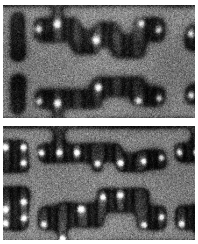


Detected Vias: 2

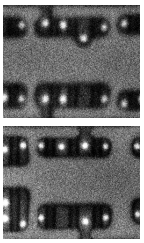
Detected True Positives: Functional Standard Cell Replacements.



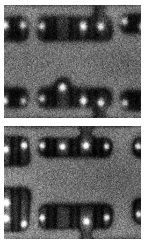
Difference Score: 0.271
Without via mask



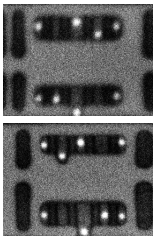
Difference Score: 0.249
Without via mask



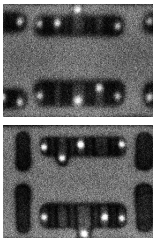
Difference Score: 0.04
With via mask



Difference Score: 0.051
With via mask



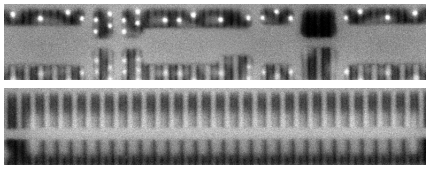
Difference Score: 0.054
With via mask



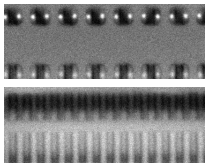
Difference Score: 0.06
With via mask

A.4. All Modifications of the 28 nm Chip

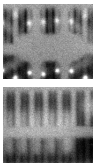
Detected True Positives: Filler Cell Replacements.



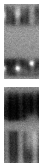
Detected Vias: 9



Detected Vias: 1

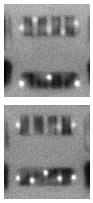


Detected Vias: 2

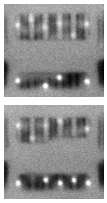


Detected Vias: 1

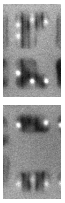
Detected True Positives: Functional Standard Cell Replacements.



Difference Score: 0.072
With via mask

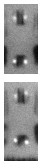


Difference Score: 0.072
With via mask

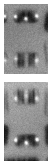


Difference Score: 0.075
With via mask

Undetected False Negatives: Functional Standard Cell Replacements.



Difference Score: 0.028
Without via mask
Difference Score: 0.019
With via mask



Difference Score: 0.06
Without via mask
Difference Score: 0.036
With via mask

Difference Score: 0.044
Without via mask
Difference Score: 0.04
With via mask