A Layout-driven Framework to Assess Vulnerability of ICs to Microprobing Attacks

Qihang Shi[1], Navid Asadizanjani[2], Domenic Forte[2], Mark M. Tehranipoor[2]
[1]ECE Department, University of Connecticut, qihang.shi@engr.uconn.edu
[2]ECE Department, University of Florida, {nasadi, dforte, tehranipoor}@ece.ufl.edu

***Abstract*—Microprobing attacks against integrated circuits (IC) for security critical applications have become a serious concern. With the help of modern circuit editing techniques, an attacker could remove layers of materials and expose wires carrying security critical information for probing. Existing protection methods use active shielding to detect such attacks. However, this technique has been proven to be ineffective, while layers of trigger wire mesh introduce prohibitive cost overhead. In this paper, we investigate the problem of protection against microprobing attacks and present a method to scan layout for microprobing vulnerabilities so that more secure and less costly protections can be developed. Exemplary applications on OpenSPARC T1 core layout is used to evaluate the proposed flow and substantiate findings.**

I. Introduction

Growing physical attacks has caused concern for design of integrated circuits for security-critical applications. Physical attacks circumvent encryption by attacking their silicon implementations. Microprobing is one kind of physical attack that directly probes at signal wires in order to extract sensitive information [1]. Successful microprobing attacks have been reported on smartcards and microcontrollers in mobile devices [15], [16]. In a successful microprobing attack, plaintexts such as personal data, code format intellectual property (IP) or even encryption keys can be compromised [2].

Most security critical ICs reinforced against microprobing attacks with active shield to detect a breach and zeroize sensitive information once a breach has been detected. However, major problems exist with this approach. Active shields are designed to cover the entirety of the die, in some designs more than one metal routing layer is required. This puts a prohibitively high cost on the design, and leaves ICs fabricated with technologies offering smaller number of available routing layers dangerously exposed to probing attacks. Furthermore, research has shown using active shields in the top metal layer of an IC to be very ineffective against microprobing attacks [16].

A. Circuit Microprobing Techniques

Circuit microprobing refers to techniques that allow an attacker to directly observe partial or full sensitive information, e.g. plaintexts or encryption keys. ICs designed for security-critical applications such as smartcards, microcontrollers in mobile devices and security tokens [15]–[17] are among the most common victims to this kind of attacks. Unfortunately, a lot of these applications also have exploitable security weaknesses [17], probably due to tight budget margins. Examples include One-Time-Programmable (OTP) memories used to store configuration and passwords rewritable with ultraviolet (UV) light, password boot-strap-loader easy to circumvent, polysilicon fuses easy to read optically and easy to rewrite, overly reused IPs that make exploits against them contagious, etc. Some of these exploits might be possible to fix with better designs; however, the disparity between technology they use due to cost and the capabilities of milling instrument of a determined attacker is less likely correctable in foreseeable future.

Microprobing attacks are categorized as invasive attacks together with fault injection and circuit editing because they all require complete removal of the package and exposure of signal routing. Wires of targeted nets that the attacker wishes to reach are likely buried under multiple passification, metal, and dielectric layers (shown in Figure 1). On ICs fabricated with feature dimensions larger than 0.35μm, laser cutters can be used to remove these layers [1]. For technologies of lower dimensions, currently the most common and powerful tool is the Focused Ion Beam (FIB) [6]. With the help of FIB, an attacker can mill with sub-micron or even nanometer level precision [21]. The most common method to protect IC from milling is the active shield, which places signal-carrying wires on top metal layers [8]–[12].The expectation is that the milling will cut off at least one of these wires and trigger the payload, which usually consists of zeroizing the sensitive information. However, in addition to milling, FIB is also capable of depositing conducting traces [20], which adds circuit editing to the attacker's repertoire. This capability allows the attacker to completely disable the active shield by editing its control circuitry or payload, if it proves too difficult to bypass [15]. Nevertheless, bypassing is still preferable for the attacker as it saves time. Deciding factor to bypass the shield is the *aspect ratio*. Aspect ratio is a measure of the FIB performance defined as the ratio between milled hole depth and diameter [3]. FIB instrument with higher aspect ratio can be expected to mill a hole of smaller diameter,which will make bypassing active shield easier. When milling in nanometer scale and applied on silicon ICs, state-of-art FIB systems can reach an aspect ratio up to 8.3 [4]. Another way to bypass the active shield is through *back-side* microprobing attacks [7], which probe at transistor activities from the silicon substrate (bottom layer in Figure 1), rather than *front-side* which probes from passification layer (top layer in Figure 1) towards metal routing layers. This is facilitated by utilizing either the phenomenon of Photon Emission (PE) or Laser Voltage Techniques (LVX) [5]. Both techniques can observe current or voltage in transistor channels, thereby deduce logic values in that transistor. These microprobing attacks are very hard to defend against since conventional IC design process doesn't place anything beneath the silicon substrate, and both methods being passive makes detection of such attacks quite impossible. However, both methods require observation of photon emissions, which makes them limited by the wavelength of emitted photons. As technology node advances and feature size shrinks, emissions from more devices will become indistinguishable, thus making microprobing attacks from back-side difficult [5].

B. Protective Designs against Microprobing Attacks

To protect against microprobing attacks, two categories of techniques exist: techniques that stops microprobing, and techniques that make it impossible for information gained from microprobing to become useful to an unauthorized user.

Existing techniques designed to stop microprobing usually perform their duty by detecting and then zeroizing sensitive information. This can be achieved either by detecting the actual activity of microprobing or activities essential for microprobing to work. The more widely studied and attempted approach is to detect hardware tampering by building a mesh of trigger wires to cover the design [8]–[12].This is called an active shield, because the trigger wires are supposed to be constantly monitored in order to detect an attack. Some shield designs are analog: for example, the authors in [8] use capacitance measurement to detect damage done to it, and thereby detect tampering. The problem with analog shield designs is that analog sensors rely on parametric measurement, which has been shown to be weaker [15]. Therefore,

* This work was sponsored in part by AFOSR MURI grant under award number FA9550-14-1-0351.

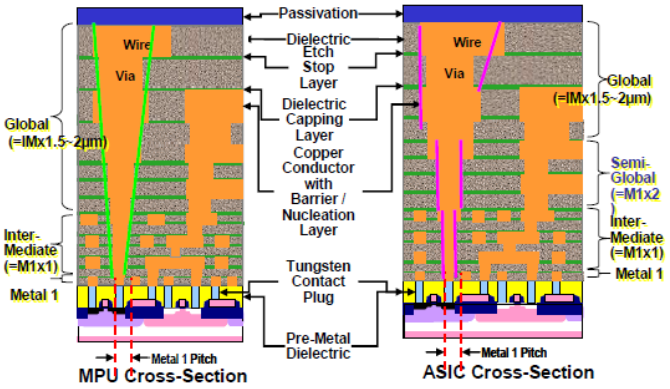


Fig. 1: Typical cross-sections of microprocessors (MPU) and Application-Specific Integrated Circuits (ASIC) [19]

digital active shields have been proposed [10]–[12]. These methods send digital random vectors through the trigger wires, and check whether received vectors are altered. A milling through the mesh would be reliably detected when it cuts off at least one of the trigger wires. The authors in [11] investigated the problem of possible *prediction attack*, where an attacker could predict the next random vector to be sent if the random vector generation is not secure enough. The authors then presented a design where block ciphers in Cipher Block Chaining (CBC) mode are used to generate secure random vectors. Another research proposed to obfuscate layout routing of the active shield so that the attacker would not be able to figure out how to perform a successful rerouting attack [12].

One problem with constructing an active shield is routing overhead. Authors in [13] presented a design to detect act of probing by monitoring change of capacitance on security critical nets, as a cheaper alternative to the more popular active shield method as it requires far less area and routing overhead. In addition to hardware based approaches, one cryptographical method called t-private circuits [14] proposed to modify the security-critical circuit so that at least $t + 1$ probes are required by an attacker to extract one bit of information.

C. Motivation

Even though back-side attacks have been proposed, front-side attacks is still worth investigating due to photon wavelength limitation, and that security critical designs may choose to fabricate a *back-to-back* 3D IC to avoid leaving back-side exposed [11]. Therefore, protection against front-side attacks remains important for antiprobing designs.

Among existing protection methods against front-side attacks, active shield remains the most-investigated method. However, no existing literature has investigated the question whether the top routing layers are the best place to detect breach. In fact, top routing layers are known to have much larger minimum wire widths [18], making it less protective than lower layers. This is especially true for devices such as smartcard, which are often fabricated with technology of larger dimensions such as 350 or 600 nm [16]. Another problem with active shield method is at least an entire metal routing layer must be dedicated to the shield. This does not go well with designs with tight cost margin, or designs with few routing layers. A lot of ICs that will likely fall victim such as smartcard [16] or microcontrollers in distributed security applications [15] neither have a very wide cost margin nor a lot of routing layers. And finally, microprobing with FIB can escalate to circuit editing. It would also be unrealistic to assume that the attacker would stop at only extracting information, without injecting any of his own. A detect-zeroize approach difficult to bypass will likely encourage the attacker to disable it. In practice, FIB has been shown capable of this [15].

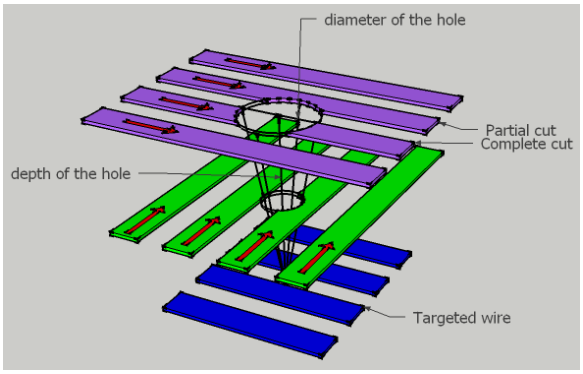


Fig. 2: Assumptions on FIB-based milling undertaken in this study.

These problems suggest that there is no "magic bullet" in antiprobing designs. A more realistic approach is to create a framework to evaluate protection designs in terms of their performance against known exploits, and provide mathematical guidance in layout design so that vulnerabilities to probing can be reduced. In this paper, we present these following contributions:

- A layout-driven framework to assess designs against microprobing attacks considering known attacks and exploits.
- A mathematical analysis on bypassing shields with FIB at any angle.
- A verification algorithm based on a mainstream layout editor (Synopsys IC compiler) to quantitatively evaluate a post-place-and-route design in terms of exposed area vulnerable to microprobing by security-critical nets; To our knowledge, the proposed algorithm is the first to provide this capability.
- Investigation of protection design issues with presented verification algorithm on OpenSPARC T1 core.

The rest of this paper is organized as follows. Section II presents a mathematical formulation of the microprobing problem from the layout design point of view. Section III presents the proposed framework, and the flow of the verification algorithm. In Section IV, we give sample evaluation results on proposed algorithm performance as well as protection design issues as laid out in the framework using OpenSPARC T1 core, before concluding the paper in Section V.

II. Microprobing Problem

In this study, we consider a milling scenario using FIB technology as shown in Figure 2, where colored bars are used to represent metal wires on different routing layers. For the sake of argument, assume lowest wires in the figure are on layer n, green on layer $n + p$, top wires on layer $n + q$, and the attacker wishes to probe at one of the wires on layer n to extract sensitive information. The hollowed-out cone shown in the figure represents a hole milled with FIB equipment. In reality, a milling hole for the purpose of microprobing will probably be larger for the probe tip to maintain a reliable connection, and what Figure 2 showed is a best-case scenario for the attacker and worst-case scenario for the designer.

From a layout point of view, active shield designers are interested in the scenario where the attacker would make a mistake and completely cut off one metal wire at purple layer, for the purpose of detecting the attacker with a difficult-to-mistake event. It is possible that a partially cut wire may be detected by its impact on circuit timing, similar to the analog shield idea [8]; However due to reliance on afore-mentioned weakness due to reliance on parametric measurement, we have yet to see a digital active shield proposed to do this. Therefore in this study we focus on detection method based on complete cuts only.

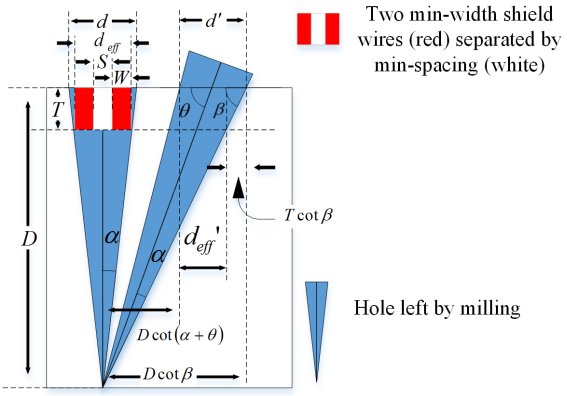


Fig. 3: Geometric calculations for non-perpendicular milling scenario.

One known exploit on active shields is to create a reroute between identified equipotential points by circuit editing with FIB, so that the net would not become open when sections of the wires are removed [16]. This forces active shield designs to only use parallel wires of minimum spacing and widths [11]. In this case, the center of the hole least likely to result in a complete cut of a wire is in the center of the space between any two wires. Conversely, the designer need to ensure within $d_{\text{eff}} = 2W + S$ the hole is at least as deep as $T = (A/R)W$, where W and S are shield-layer metal widths and minimum wire spacing, and (A/R) is the aspect ratio of the wire. This creates a restriction of milling hole diameter d on active shield layer

$$\begin{aligned} d &\leq d_{\text{eff}} + \frac{1}{R_{\text{FIB}}}T \\ &= 2W + S + \frac{W}{R_{\text{FIB}}} \end{aligned} \tag{1}$$

must be satisfied or wires will be cut, where R_{FIB} is the maximum aspect ratio of FIB. If we take $W = S$ (as ITRS did [19]), Equation 1 further simplifies into $d \leq (3 + \frac{1}{R_{\text{FIB}}})W$.

One interesting question is whether attacker would benefit if instead of milling vertically, he mill at an angle, as shown in Figure 3. If we assume he was able to mill at $\theta \leq \frac{1}{2}\pi$, then he will cut off wires within region d'_{eff} instead of d_{eff}

$$\begin{aligned} d'_{\text{eff}} &= d' - \begin{cases} T\cot\beta & , \theta \in [0, \frac{1}{2}\pi - \alpha] \\ T(\cot\beta - \cot(\theta+\alpha)) & , \theta \in [\frac{1}{2}\pi - \alpha, \frac{1}{2}\pi] \end{cases} \\ &= \frac{\sin 2\alpha}{\sin(\theta+\alpha)\sin(\theta-\alpha)}D - \\ &\begin{cases} T\cot(\theta-\alpha) & , \theta \in [0, \frac{1}{2}\pi - \alpha] \\ T(\cot(\theta-\alpha) - \cot(\theta+\alpha)) & , \theta \in [\frac{1}{2}\pi - \alpha, \frac{1}{2}\pi] \end{cases} \end{aligned} \tag{2}$$

Taking derivative of $\frac{d'_{\text{eff}}}{d_{\text{eff}}}$ and letting it equal to zero yields minimum point at

$$\begin{aligned} \theta_0 &= \frac{1}{2}\text{acos}\left(\frac{bc - \sqrt{b^2c^2 - (a^2+b^2)(c^2-a^2)}}{a^2+b^2}\right), \text{ where} \\ a &= (2(A/R)\tan\alpha + 6)\sin 2\alpha \\ b &= 2(A/R)\tan\alpha\cos 2\alpha \\ c &= 2(A/R)\tan\alpha \end{aligned} \tag{3}$$

If we further assume $(A/R) = 2.5$ as in [18] (ITRS uses 2.34 [19]), Equation 3 yields the following reduction in d'_{eff} over d_{eff} shown in Table I. From the table we see that by milling at approximately $68° - 69°$ angle the attacker can effectively reduce the diameter of area by $8 - 12\%$, making it easier to bypass the shield. Since bypassing the shield is considered a convenient and preferable approach [15], this possibility makes FIB even more lethal for shields with wide top layer wires.

TABLE I: Maximum achievable reduction of d_{eff} by milling at an angle.

R_{FIB}	5	6	7	8	9	10
$\frac{d'_{\text{eff}}}{d_{\text{eff}}}$ (%)	92.12	90.58	89.47	88.63	87.98	87.45
θ_0 (°)	68.93	68.69	68.52	68.38	68.28	68.19

III. Proposed Framework and Layout Verification Algorithm

A. Layout-Driven Framework to Assess Antiprobing Designs

Before presenting the framework to assess protection designs against microprobing attacks, it is essential to establish the principles of these designs. Otherwise, research could become sidetracked by objectives unnecessary or insufficient, and assessment would lack a standard for comparison.

One pitfall for the designer might be to underestimate the capability of the attacker. When considering tools available to a microprobing attack, it is important to remember that attackers capable of nano-meter scale milling is not restricted to microprobing alone. FIB itself allows circuit editing, which enables attacker to disable the whole shield by tying its detection bit to ground. Lasers can be used to inject arbitrary values to confuse protective mechanism. Indeed, both techniques have been reported successful [15]. As a result, while designs that can defeat all known attacks might not be impossible, it certainly is impractical to pursue for most devices.

Meanwhile, another myth is to underestimate the difficulty of microprobing attack. It is important to remember that attackers are likely to find a way in does not mean protection design is futile. The goal of a microprobing attack is sensitive information, and sensitivity decays with time. Information expires. Passwords are rotated. Backdoors are fixed with security updates. Even functional designs are phased out of market by new generations. Therefore, if delayed long enough, objectives of even an attacker with infinite resources can be denied.

In addition to delaying the most well-equipped attackers, it is also in the interest of the designer to deter less well-equipped attackers. This is especially true for low-cost devices such as security tokens and smartcards. This deterrence can be performed in terms of capability or information. Countermeasures vulnerable to the most cutting edge instruments might still filter out attackers that do not have access to such capabilities, and using custom designs instead of IPs reduce the risk of having a vulnerability when an IP you use is successfully attacked.

In addition to the aforementioned principles, a protection design should always be assessed with knowledge of the attack it is designed to prevent. Published microprobing attacks [15] consist of these following fundamental steps, each must be successful for the attack to succeed:

- Reverse engineer a sacrificial device to get its layout and find target wires to microprobe;
- Locate the target wires with milling tool;
- Reach the target wires without damaging target information;
- Extract target information.

Each step can have a number of alternative techniques where success with only one of them is necessary. For example, locating target wires in layout can be done by reverse engineering the design or with information from a similar design. Obfuscation can force the attacker to spend more time on this step, but if the IP is reused in another design it would allow attacker to circumvent it.

Based on principles above we propose the following framework to assess a design for vulnerability to microprobing attacks:

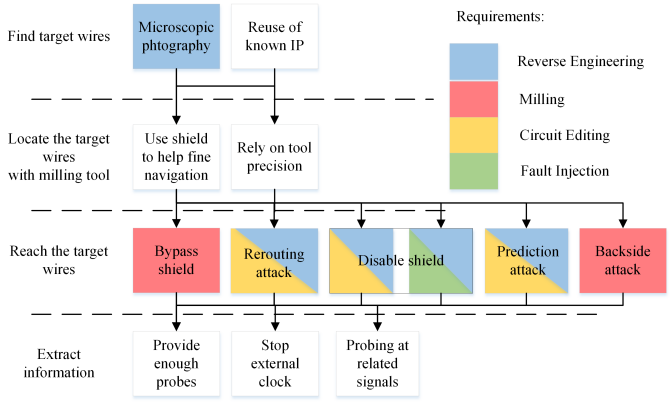


Fig. 4: Diagram of known microprobing techniques for assessment of design vulnerability.

- For each necessary step during a microprobing attack, enumerate all known alternative techniques, the capability required by this technique, whether and how much does the design change the expected time cost on each technique;
- the protection against attackers with infinite resources is represented with the sum of techniques with the lowest time cost from each necessary step;
- the protection against less well-equipped attackers can be assessed by repeating the same process without techniques requiring unavailable capabilities.

In this framework it is possible for a particular microprobing technique to have an infinite time cost against a particular design: for example, an active shield with wires too thin for current FIB to bypass. However, the overall time cost is unlikely to be infinite due to existence of very powerful techniques such as circuit editing: in the aforementioned example, the attacker could opt to remove the shield and disable it by fault injection or circuit editing at shield control or payload circuitry, a technique known as *disabling shield* [15]. To better illustrate this, we provide a diagram of known microprobing techniques [1], [15]–[17] in Figure 4 as an example.

Shown in Figure 4 is a typical flow of a microprobing attack, where each step is shown in a row and each block shows an alternative technique to complete that step. Some techniques are shaded with colors to represent the particular capability to enable that technique. *Disable shield* technique is shown with two blocks with blue triangles to show it can be completed either with circuit editing or fault injection, but in both options reverse engineering is required. Techniques in white boxes that do not have a colored alternative show possible exploits from avoidable design flaw rather than lack of protection. For example, "Use shield to help fine navigation" is possible if shield wires were not placed in 45° with regard to functional routing [15]; and if no internal clock source is used, attacker could simply "stop external clock" to extract all information without having to use multiple probes. Based on these known microprobing techniques we may assess the protection of a few published designs, as shown in Table II. From the proposed framework we can see that layout is of central importance in both restricting the attacker's options and increasing his time cost. If area exposed to milling can be conveniently found, it will enable designers to create antiprobing designs with better all-around resilience. For this purpose, we present an algorithm to evaluate and find *exposed area*.

B. Algorithm to find exposed area

First, consider the wires as shown in Figure 2. An active shield will need a complete cut to detect milling. Using similar calculation as in

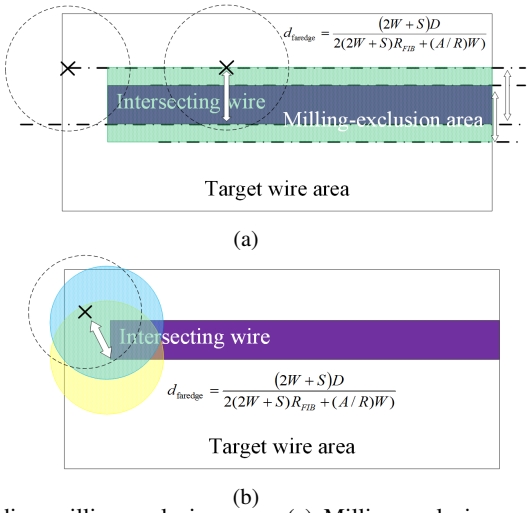


Fig. 5: Finding milling-exclusion area. (a) Milling-exclusion area on sides of intersecting wire; (b) Milling-exclusion area on ends of intersecting wire.

Section II, complete cut will happen if center of milling exists within d_{faredge} from the far edge of the wire, where

$$d_{\text{faredge}} = \frac{(2W+S)D}{2(2W+S)R_{\text{FIB}} + (A/R)W)} \tag{4}$$

where d is the diameter of the hole, D is the depth of the hole, (A/R) is the aspect ratio of the shield wire metal, and $R_{\text{aspect ratio}}$ is the aspect ratio given by the FIB technology the attacker is using. The aspect ratio represents the best FIB the shield will be able to defend against.

Equation 4 shows possibility to find the area which milling center should not fall inside. We term this area the *milling-exclusion area*. The desired *exposed area* will be its complement. Figure 5 shows how this area can be found for any given target wire and a wire on a higher layer capable of projecting this milling-exclusion area for it (henceforth termed as *intersecting wire*), assuming both are rectangular.

Boundaries of the milling-exclusion area can be found in two possible cases for a rectangular intersecting wire: the boundaries on the sides of the intersecting wire, and at both ends. The first kind is quite intuitive. As shown in Figure 5a, the center of the milling cannot fall within d_{faredge} from the farther edge of the intersecting wire, therefore boundaries of the first kind are two straight lines, each d_{faredge} away from the farther edge. The other kind of boundaries on ends are a bit more complex. Let's look at Figure 5b. Consider the milling hole marked by the dotted circle. For it to precisely cut off the intersecting wire at each corner of the intersecting wire, its center must be on the edge of another circle centered at that corner, with same radius as itself. Any point within that other circle will still cut off that corner, although not necessarily the other corner. Therefore, the intersection area of both circles centered at both corners at an end constitute the complete set of

TABLE II: Performance against known microprobing techniques of published designs

Designs	Protection against				
	Bypass Shield	Rerouting Attack	Disable Shield Backside Attack	Prediction Attack	Related Signals
Analog Shield	Weak [15]	No	No	N/A	Yes
Random Active Shield [12]	Yes	Yes	No	No	Yes
Cryptographically Secure Shield [11]	Yes	Yes	No	Yes	Yes
PAD [13]	N/A	N/A	No	N/A	No

milling center locations that will guarantee cut of both corners, i.e. a complete cut. Consequently, any intersecting wire rectangular in shape will project a milling-exclusion area whose shape is the union of the shape shown in Figure 5a and Figure 5b.

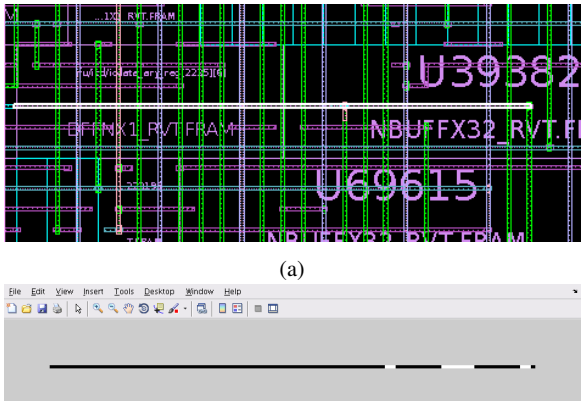
Now, wires in layout designs are seldom rectangular, but they are always consisted of a number of rectangular wires, usually called *shapes* by layout design tools. By iterating through each of these constituent rectangular wires, mill-exclusion areas from each intersecting wire can be projected onto each wire that may carry sensitive information and become target of microprobing attack. This process is elaborated in the pseudocode as shown in Algorithm 1.

Input: *targeted_nets, precision, all_layers*
Output: *draw.script*
begin
  targeted_wire_shapes ⇐ get_net_shapes(*targeted_nets*)
  N ⇐ sizeof_collection(*targeted_wire_shapes*)
  for $(i = 1 : N)$ **do**
    targeted_wire_shape ⇐ *targeted_wire_shapes*(i)
    canvas_size ⇐ get_sizes(get_bounding_box(*targeted_wire_shape*))*precision
    Print command in *draw.script* to create canvas in *draw.script* whose size equals to *canvas_size*
    layers_above ⇐ get_layers_above(*all_layers*, get_layerof(*targeted_wire_shape*))
    M ⇐ sizeof_collection(*layers_above*)
    for $(j = 1 : M)$ **do**
      this_layer ⇐ *layers_above*(j)
      d_faredge_on_thislayer ⇐ $\frac{(2W+S)D}{2(2W+S)R\_\text{FIB}+(A/R)W)}$
      intersecting_wire_shapes ⇐ get_net_shapes(*targeted_nets*) in get_bounding_box(*targeted_wire_shape*) on *this_layer*
      L ⇐ sizeof_collection(*intersecting_wire_shapes*)
      for $(k = 1 : L)$ **do**
        intersecting_wire_shape ⇐ *intersecting_wire_shapes*(k)
        Print command in *draw.script* to create projection in *draw.script* whose radius/widths equals to *d_faredge_on_thislayer*
      end
    end
  end
end

Algorithm 1: Proposed locator algorithm for exposed area.

As shown in Algorithm 1, the proposed methodology starts with a set of logic nets. The algorithm first identifies their constituting wire shapes in *targeted_wire_shapes*. For each targeted wire shapes, a bitmap canvas is created, onto which mill-exclusion areas are to be projected once found. These coordinates are also given to the layout design tool to find *intersecting wire shapes* on each layer above. For each layer, a different d_{faredge} is calculated, which is then used for projections from all intersecting wire shapes on that layer. Coordinates of each intersecting wire shape are also retrieved to compute its mill-exclusion area, which is then projected to the aforementioned canvas (results shown in Figure 6). Projection is done by locating ends and sides of each intersecting wire shape and print the corresponding projected mill-exclusion areas. After all mill-exclusion areas are projected, running the resulting script *draw.script* can easily determine existence and area of exposed area.

For processing efficiency and adaptability, both canvas creation and projection steps are stored by the layout design tool part of the algorithm in the format of MATLAB scripts. Considerations of microprobing attacks at non-perpendicular angles can also be included with simple modifications with trigonometric functions. Another possible concern is the precision of the bitmap method. The proposed algorithm rounds toward minus infinity on borders, i.e. errs towards false positive. However, since mill-exclusion areas are convex, overlapping of mill-exclusion areas would unlikely cause the algorithm to declare a vulnerable point when there is none either.



(a)

(b)

Fig. 6: Exemplary results produced by proposed algorithm. (a) Exemplary targeted wire (highlighted) in layout; (b) Mill-exclusion area (black) projected on canvas of same wire.

IV. Evaluation Results

A. Evaluation of algorithm

In this section, we present an evaluation on the proposed algorithm. The objective is to find out how efficient can the proposed algorithm be and how much area in a typical unprotected design is exposed to microprobing attacks. For this purpose, layout of an OpenSPARC T1 core using Synopsys SAED 32nm technology library is chosen for the algorithm to inspect. For the purpose of verification, two groups of nets are selected to serve as targeted wires: first we look for long wires in the design, then we evaluate on wires on lower layers. Long wires are chosen for they resemble data buses, which are typical targets for microprobing attacks. Wires routed in lower layers are less exposed than wires routed in higher layers and therefore forcing nets that could carry security-critical information to route on lower layers can be a sensible alternative to active shield. For this evaluation we used a resolution of 10nm, and we assume the maximum $R_{\text{FIB}} = 10$.

The long wires in this evaluation are picked based on the diagonal length of the smallest rectangle encompassing all of its shapes. All long wires thus picked have a diagonal length of at least 500μm, a number chosen to be longer than 99% of all signal route nets. On the other hand, nets routed on lower layers are restricted to not have shapes on layers higher than metal-4. This layer is picked because it is likely realistically possible if the designer tries to push his more vulnerable nets into lower layers. 5000 nets in lower-layer group of nets and 128 nets in long-wire group of nets are investigated. Their running time and exposed area are shown in Table III.

TABLE III: Evaluation results on long nets and nets on low layers

Performance	Nets on Metal-4 or Lower Layers	Long Nets
Total Number of Nets	5000	128
Total Processing Time (s)	27145	11708
Processing Time per Unit Area ($s/\mu m^2$)	5.1242	2.1297
Total Area (μm^2)	5320.58	5497.66
Exposed Area (μm^2)	4339.84	4869.21

In both cases, the proposed algorithm was able to finish processing within a few hours for a few thousand μm^2. The speed could definitely be further improved, but it seems acceptable for practical purposes, especially if we consider that the number of probes an attacker can simultaneously support is also restricted [14]. Despite only having 128 nets, a much smaller number compared to 5000 nets in lower-layer group, the long-wire nets almost have the same total area. It shows a greatly reduced difficulty for the attacker to attempt microprobing at

TABLE IV: Evaluation of active shield performance

Performance \ R_{FIB}	5	6	7	8	9	10
% shield ineffective (%)	1.52	3.82	19.50	45.90	100	100
Exposed Area (μm^2)	4364.63	4507.47	4656.88	4760.98	4869.21	4869.21

these long wires than at wires carrying signals related to it but otherwise much shorter and on lower layers. This could suggest that having those related signals might not be as mortal a sin as it first appeared. Indeed, judging from the difference in percentage of exposed area between the two groups of nets (81.57% in lower-layer group, 88.57% in long-wire group), long wires are more exposed than wires on lower layers.

B. Performance evaluation of active shield

This evaluation investigates the protection performance of active shield against FIB-based microprobing attack. Using the same layout, in this evaluation we assume that on the topmost MRDL layer, horizontal active shield wires are present. Wire width and wire spacing of this shield are both assumed to equal to 2μm, as was given by minimum wire width and minimum spacing of that layer in the technology file. The targeted nets are the same nets as the group of long nets in Subsection IV-A. Results are given in Table IV. In Table IV, the row "shield ineffective" indicates how many wire shapes among the total cannot benefit from the coverage of the shield at all. Note that this is based on number of shapes without regard to their area, and over 80% of these shapes are below metal layer 4. From the results we can see even on very low R_{FIB}, the long wires are not benefiting much from the shield. This result substantiated our earlier observation that current entire-layer active shield are restricted by very wide top layer metal wire width since they cannot be placed on lower layers without making all layers above it unavailable to the design. Compared to results we see in Table III, it makes sense to try using functional signal routes instead.

C. Future research directions

To our knowledge, the proposed algorithm provides the first quantifiable way to verify and evaluate microprobing vulnerabilities. This will open up a number of new opportunities in protection designs. With proposed algorithm, active shield no longer need to cover an entire layer to ensure security, therefore it can be relocated to better performing layers to improve FIB aspect ratio it can protect against. Weak links in the design, such as control and payload wires, could be buried with functional signal routes and made more resilient to attacks. Covering with multiple signal routes leads to greatly elevated requirement of reverse engineering and consequently time cost for the attacker, since he has to ensure the information gained is unspoiled and has no way to verify it. This approach can be promising if used in conjunction with anti-reverse engineering designs, as the latter greatly increase time cost in reverse engineering [6]. This could also allow protection to designs too tight in cost margin or number of layers to afford an entire layer for active shield. For this purpose, more layout-based tools could be developed to identify security critical nets, find functional nets most suitable to serve as intersecting wire shapes, exploit faster probing assessment metrics that can integrate into existing layout optimization flow, etc.

V. Conclusion

Methods to reinforce IC in security critical applications against microprobing attacks under active research interest are plagued with high cost, weaknesses that could be exploited by attackers, and incompatibility to technologies with few layers. In this article, we present a layout-driven framework to assess designs for vulnerabilities to microprobing attacks. Based on design principles and assessment metrics we have established, We presented an algorithm to analyze layout designs for potential vulnerabilities to microprobing attacks, and evaluated its performance on the layout of an OpenSPARC T1 core. Evaluation shows its potential to process large amount of nets with practical time cost. We expect it to serve as a basis for future methodologies to protect against microprobing attacks that are more effective, require lower hardware cost and applicable to a wider variety of ICs.

References

[1] Skorobogatov, S., "Physical attacks on tamper resistance: progress and lessons," Proc. of 2nd ARO Special Workshop on Hardware Assurance, Washington, DC., 2011

[2] Anderson, R., "Security engineering: A guide to building dependable distributed systems," Wiley, 2001.

[3] Fu, Y.; Ngoi, K. A. B., "Investigation of aspect ratio of hole drilling from micro to nanoscale via focused ion beam fine milling," 2005

[4] Wu, H.; Ferranti, D.; Stern, L., "Precise nanofabrication with multiple ion beams for advanced circuit edit," in Microelectronics Reliability, vol. 54, iss. 910, pp. 1779-1784, September-October 2014

[5] Boit, C.; Helfmeier, C.; Kerst, U., "Security Risks Posed by Modern IC Debug and Diagnosis Tools," in Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on, IEEE, pp. 3-11, August 2013

[6] Quadir, S. E.; Chen, J.; Forte, D.; Asadizanjani, N.; Shahbazmohamadi, S.; Wang, L.; Chandy, J.; Tehranipoor, M., "A Survey on Chip to System Reverse Engineering," to appear ACM Journal on Emerging Technologies in Computing Systems (JETC).

[7] Helfmeier, C.; Nedospasov, D.; Tarnovsky, C.; Krissler, J. S.; Boit, C.; Seifert, J. P., "Breaking and entering through the silicon," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 733-744, ACM, November 2013

[8] Laackmann, P.; Taddiken, H., "Apparatus for protecting an integrated circuit formed in a substrate and method for protecting the circuit against reverse engineering," U.S. Patent No. 6,798,234. 28 September 2004

[9] Ling, M.; Wu, L.; Li, X.; Zhang, X.; Hou, J.; Wang, Y., "Design of Monitor and Protect Circuits against FIB Attack on Chip Security," in Computational Intelligence and Security (CIS), 2012 Eighth International Conference on , pp.530-533, 17-18 November 2012

[10] Beit-Grogger, A.; Riegebauer, J., "Integrated circuit having an active shield," U.S. Patent No. 6,962,294. 8 November 2005

[11] Cioranesco, J.-M.; Danger, J.-L.; Graba, T.; Guilley, S.; Mathieu, Y.; Naccache, D.; Xuan Thuy Ngo, "Cryptographically secure shields," in Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on , vol., no., pp.25-31, 6-7 May 2014

[12] Briais, S.; Cioranesco, J.-M.; Danger, J.-L.; Guilley, S.; Naccache, D.; Porteboeuf, T., "Random Active Shield," in Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012 Workshop on , pp.103-113, 9-9 September 2012

[13] Manich, S.; Wamser, M.S.; Sigl, G., "Detection of probing attempts in secure ICs," in Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on, pp.134-139, 3-4 June 2012

[14] Ishai, Y.; Sahai, A.; Wagner, D., "Private circuits: Securing hardware against probing attacks," Advances in Cryptology-CRYPTO 2003. Springer Berlin Heidelberg, 2003. 463-481.

[15] Ray V., "FREUD Applications of FIB: Invasive FIB Attacks and Countermeasures in Hardware Security Devices", East-Coast Focused Ion Beam User Group Meeting, Feburuary 2009

[16] Tarnovsky C., "Tarnovsky Deconstruct Processor," Youtube, https://www.youtube.com/watch?v=w7PT0nrK2BE, 2013

[17] Tarnovsky C., "Security Failures In Secure Devices", Black Hat Briefings, Feburuary 2008

[18] FreePDK45: Metal Layers. http://www.eda.ncsu.edu/wiki/FreePDK45:Metal_Layers

[19] International Technology Roadmap for Semiconductors 2013 Edition. http://www.itrs.net/ITRS%201999-2014%20Mtgs,%20Presentations%20&%20Links/2013ITRS/Home2013.htm

[20] Wu, H.; L. Stern; D. Xia; D. Ferranti; B. Thompson; K. Klein; C. Gonzalez;P. Rack, "Focused Helium Ion Beam Deposited Low Resistivity Cobalt Metal Lines with 10 nm Resolution: Implications for Advanced Circuit Editing," Journal of Materials Science: Materials in Electronics 25 (2): 587-595, 2014

[21] Sidorkin, V.; van Veldhoven, E.; van der Drift, E.; Alkemade, P.; Salemink, H.; Maas, D., "Sub-10-nm nanolithography with a scanning helium beam," Journal of Vacuum Science & Technology B, 27, L18-L20, 2009