Chapter 5
Analyzing Circuit Layout to Probing Attack

Qihang Shi, Domenic Forte, and Mark M. Tehranipoor

5.1 Introduction

Physical attacks have caused growing concern for design of integrated circuits used in security critical applications. Physical attacks circumvent encryption by attacking their silicon implementations. IC probing is one form of physical attack that allows an attacker to access security critical information by directly accessing physical wires in the IC that carry such information [1]. For convenience, we henceforth refer to such physical wires that probing attacks target as *targeted wires*. Successful probing attacks have been reported on smartcards and microcontrollers in mobile devices [2, 3]. In a successful probing attack, plaintexts such as personal data, software-form IP, or even encryption keys can be compromised [4].

IC probing attacks happen for a spectrum of different reasons in the wild. Probing attacks circumvent encryption without rendering the targeted device inoperable, and therefore they enable unauthorized access on systems carrying sensitive information. An otherwise innocuous tech-savvy kid might want to hack his satellite TV smartcard to "unlock" channels he isn't supposed to watch [5]; The issue becomes less innocuous when authorities seize multi-million dollars' worth of asset in connection with piracy lawsuits [6]. As modern lives become more integrated with the Internet through mobile hardware, security of private data, and keys stored on these devices also raised concern. If leaked, such information can be used to further identity theft, blackmail, or a number of other threats to individual rights and liberties. The worst-case scenario is probably for systems that enable access

Q. Shi (✉)
ECE Department, University of Connecticut, Storrs, CT, USA
e-mail: qihang.shi@engr.uconn.edu

D. Forte • M.M. Tehranipoor
ECE Department, University of Florida, Gainesville, FL, USA
e-mail: dforte@ece.ufl.edu; tehranipoor@ece.ufl.edu

P. Mishra et al. (eds.), *Hardware IP Security and Trust*,
DOI 10.1007/978-3-319-49025-0_5

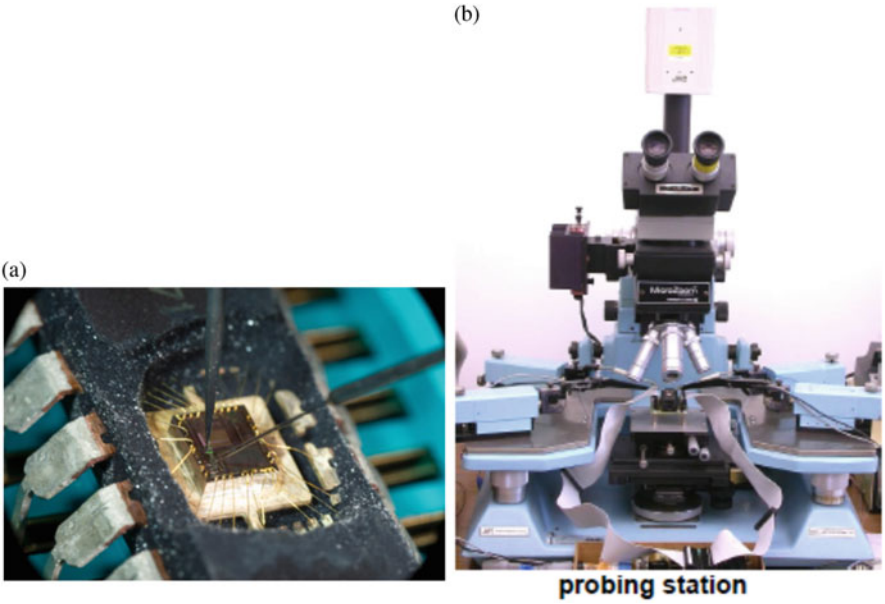

Fig. 5.1 Example of a device under probing and probing station. (**a**) An IC under probing, with package partially removed and probing pin inserted [7]. (**b**) A probing station [1]

to information related to public security—for example, the security tokens used in defense industries—to become compromised. Therefore, it follows that tamper resistance has become a default requirement in US defense acquisitions [8].

A typical probing attack involves (at least partially) remove package of the targeted device (Fig. 5.1a), place it on a probing station (Fig. 5.1), and position the metal probe to form an electrical contact with targeted wires (also shown in Fig. 5.1a). In real attacks, targeted wires are usually found by reverse engineering a sacrificial device of the same design. Reverse engineering can be a lengthy process, however, it can be accelerated if the IC under attack reuses compromised hardware IP blocks, i.e., hardware IP blocks that have been reverse engineered in previous attacks. In most IC designs, targeted wires are often buried beneath layers of passivation, dielectric and other metal layers. As an example, consider a scenario where an attacker targets a wire routed no higher than Metal 4 layer, in an IC that has the same cross-section as in Fig. 5.2. To initiate a probing attack, an attacker has to expose targeted wires either by milling through all layers above them, or the silicon substrate beneath them. In our example, he will have to mill either from passivation layer down to Metal 4, or from substrate up to Metal 4. In either scenario, use of milling tools is necessary.

Most security critical ICs are reinforced against probing attacks with active shield. This approach functions by detecting milling events and sending alarms once a breach has been detected. Here "sending alarm" is a general term referring

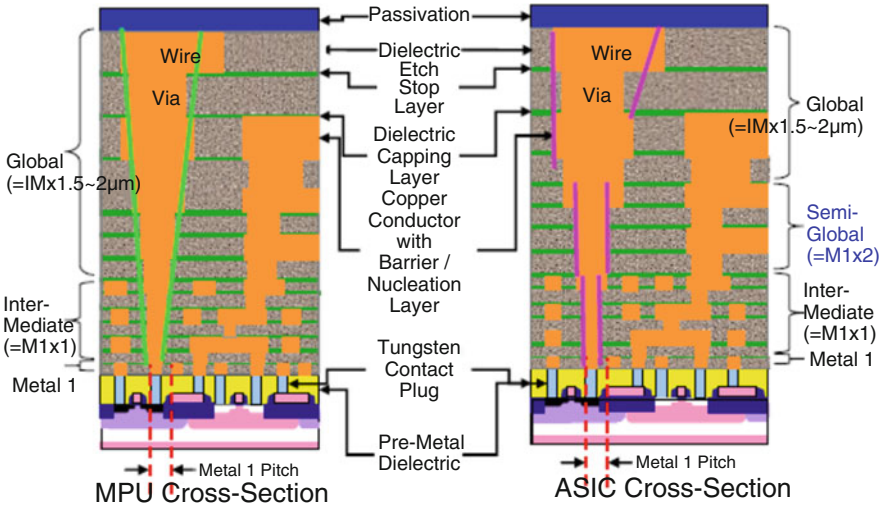

Fig. 5.2 Typical cross-sections of microprocessors (MPU) and Application-Specific Integrated Circuits (ASIC) [9]

to all proper security actions to address a probing attempt, such as zeroizing all sensitive information. Due to their popularity in research and practice, attacks and antiprobing designers often focus their efforts on discovering and improving exploits and countermeasures to penetrate or prevent penetration of the active shield.

In this chapter, we first review technologies and techniques known for their use in probing attacks. These include technologies developed for other purposes such as IC failure analysis, and specifically probing techniques such as back-side attacks. These will be covered in Sect. 5.2. Based on this knowledge, we then investigate approaches to secure designs against probing attacks, commonly known as *antiprobing* designs. This is covered in Sect. 5.3, which introduces published proposals to secure the design against probing attacks, their known problems and research to address these problems. Based on background knowledge provided in these two sections, it becomes apparent to us that evaluation of designs in terms of their vulnerabilities to probing attacks would likely prove a valuable contribution to the field. Therefore, in Sect. 5.4 we present a layout-driven framework to assess designs' vulnerabilities to probing attacks, rules of said assessment framework, a set of assumptions on state-of-the-art antiprobing designs, and an algorithm to quantitatively evaluate design for exposure to probing attacks. Finally, the chapter is concluded in Sect. 5.5.

5.2 Microprobing Attack Techniques

In this section we introduce various techniques used in probing attacks. All probing attacks will involve milling to expose targets for the probe to access; however, it is also important to acknowledge that a probing attack is a concerted effort that involves different techniques at various stages, and many techniques other than milling help shape the threat and limitations of the probing attack. Understanding of these techniques is essential to the understanding of principles when designing against such attacks.

5.2.1 Essential Steps in a Probing Attack

Before introducing specific techniques for probing attacks, let's first take a look at the *modus operandi* of a typical probing attack. In surveying reported attacks, we have found probing attacks require at least four essential steps [3], each must be successful for the attack to succeed (shown in Fig. 5.3 as different rows):

- 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, for example, by performing dynamic address mapping and block relocation on embedded memories [10] or by performing netlist obfuscation using modification kernel function [11]; but if the same hard IP is reused in another design both designs become vulnerable once that IP becomes compromised. The diagram here lists all the alternatives we know to exist, and new attacks can be easily integrated.

Shown in Fig. 5.3 is a typical flow of a probing 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 patterns to represent the particular capability to enable that technique. *Disable shield* technique is shown with two blocks each having two patterns 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 patterned 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 [3]; and if no internal clock source is used, attacker could simply "stop external clock" to extract all information without having to use multiple probes.

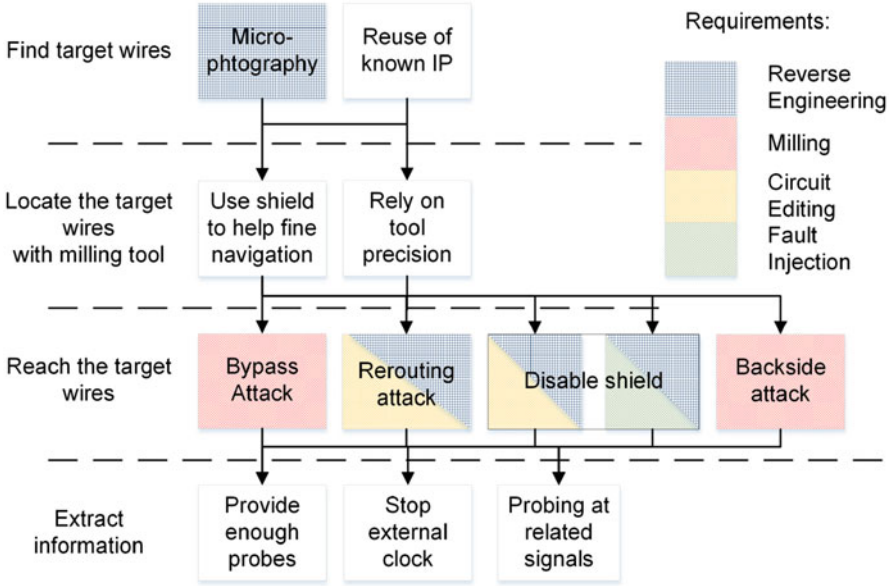

Fig. 5.3 Diagram of known probing techniques for assessment of design vulnerability

5.2.2 Microprobing Through Milling

On ICs fabricated with feature dimensions larger than 0.35 μm, laser cutters can be used to remove these layers [1]. For technologies of lower dimensions, currently the state-of-the-art tool is a technology called the Focused Ion Beam (FIB) [12]. With the help of FIB, an attacker can mill with sub-micron or even nanometer level precision [13]. Aspect ratio (see Fig. 5.4b) is a measure of the FIB performance defined as the ratio between depth D and diameter d of the milled hole [14]. FIB instruments with higher aspect ratio can be expected to mill a hole of smaller diameter on the top-most exposed layer, and therefore leave smaller impact on all the other circuitry the attacker is not interested in breaking. When milling in nanometer scale and applied on silicon ICs, state-of-the-art FIB systems can reach an aspect ratio up to 8.3 [15]. In addition to milling, FIB is also capable of depositing conducting traces [16], which adds Circuit Editing (CE) to the attacker's repertoire.

The most straightforward way to expose targeted wires through milling is to mill from the Back End of Line (BEOL), i.e., from passivation layer and top metal layer towards silicon substrate (see Fig. 5.4a). This is called a *front-side attack*. An obvious disadvantage of front-side attack is that targeted wires may be covered by other wires above it, and without thorough reverse engineering of the IC the attacker cannot know for sure whether cutting through these covering wires would corrupt the information he seeks to access. In real attacks, attackers address this problem by focusing on bus wires and try to probe from more exposed (i.e., not covered by

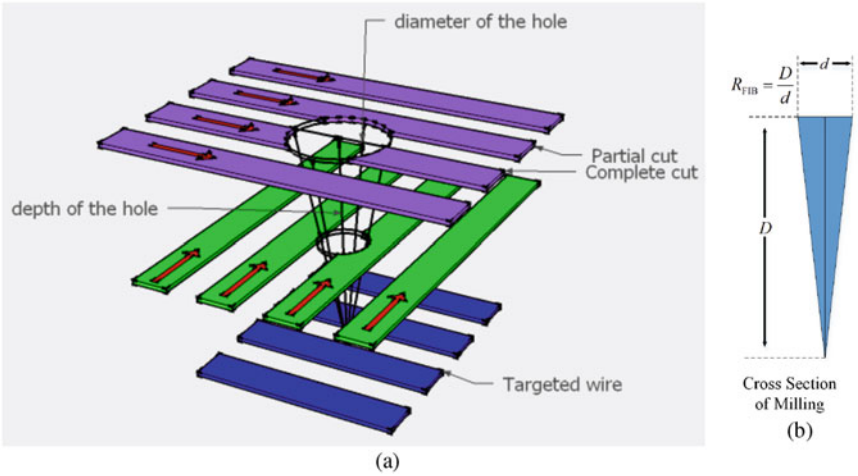

Fig. 5.4 Milling with Focused Ion Beam (FIB) technology. (**a**) Milling through covering wires to expose targeted wires. (**b**) Definition of aspect ratio of an FIB

other wires on higher layers) segments of the targeted wires [2]. The latter could be facilitated by performing reverse engineering a sacrificial device, in which case desirable milling sites can be found from exposed layout of the sacrificial device. The attacker still needs to find a way to navigate to the same sites on the target device, which might be simplified or complicated depending on whether special care was taken against this step by the designer.

5.2.3 Back-Side Techniques

In addition to milling through routing layers and inserting metal probes at targeted wires, other techniques exist. Another way is through the silicon substrate, the so-called *back-side* attacks [17]. In addition to probing at wires, back-side attacks can also access at current activities in transistor channels (bottom layer in Fig. 5.2). This is facilitated by techniques known as Photon Emission (PE) and Laser-based Electro-Optical Modulation (EOFM or Laser Voltage Techniques LVX) [18]. Between the two methods, PE can be passively observed without any external stimulation, while LVX requires IR laser illumination of the active device. On the other hand, PE manifests mainly during rise and fall time of the signal, while LVX response is linearly correlated to the voltage applied to the device. Both methods are very reliable as modern IC debug and diagnosis tools, and their legitimate uses ensure that the available tools will maintain pace with semiconductor scaling. Indeed, recent reports show that all 16 bytes encryption keys of AES-128 can be recovered in 2 h [19].

One limiting factor of the passive techniques is that both methods require observation of photon emissions, which makes them limited by the wavelength of emitted photons. For example, detection of PE on majority of instruments operate between 2 and 4 μm [20]. Depending on spatial distribution of switching activity, this might cause a problem on devices manufactured with deep sub-micron technologies. As technology node advances and feature size shrinks, emissions, especially responses from LVX techniques from more devices will become indistinguishable, thus making probing attacks from back-side difficult [18].

Another possibility of back-side attacks is circuit editing, enabled with FIB. Like with front side, wires can be cut from back-side; and in addition to cutting wires, it is also possible to deposit conductive material in holes dug into drain and source regions of transistors, serving as metal probes that can be inserted anywhere without worrying cutting open any covering wires as the front-side attack [21].

Back-side attacks are harder to defend against since conventional IC design process doesn't place anything beneath the silicon substrate. However, next generation security critical designs may choose to fabricate a *back-to-back* 3D IC to avoid leaving the silicon substrate exposed [22, 23], effectively eliminating back-side attacks all together. Therefore, protection against front-side attacks remains an important topic for antiprobing designs.

5.2.4 Other Related Techniques

This category summarizes all techniques that may be used to help further the probing attack or jeopardize a secured IC's defense against it. Some techniques can be quite essential to the task: for example, all probing attacks have to consist of some amount of reverse engineering. At least, the attacker has to locate targeted wires (or in the case of back-side attacks, targeted gates) by discovering which part of the circuit is responsible for carrying the sensitive information the attacker seeks; if cutting open a functional wire is inevitable, additional reverse engineering is required to find out cutting open which wires do not lead to corruption of the information the attacker wishes to probe. The problem with reverse engineering is the amount of work and time cost involved: since the target of probing attacks are often information that has a short life span (e.g., keys and passwords), this can sometimes be used to work against the attacker.

Some other techniques are supplemental to the main attack. One example is circuit editing. An obvious use of this technique can be used to disable the security feature, e.g., cutting off the alarm bit and rewiring it to power supply or ground. In addition to that direct approach, circuit editing could also be used to supplement other requirements, for example, clock source can be rewired to a controlled source, which can be very useful when the number of wires is too many for available probes.

5.3 Protection Against Probing Attacks

This section gives an overview of current research in securing designs against probing attacks. The focus of this section is on active shields, a protection mechanism that has received the central attention of the academia, and application on security hardware in the market. Much research has been aimed at finding shield exploits and providing fixes to stop known exploits. In addition to attack and defense with regard to active shields, we also present a mathematical model of active shield and its ability to detect milling. Other approaches that are not active shields and design details that complement active shields are also covered.

5.3.1 Active Shields

The most common method to protect IC from probing attack is the active shield, which detects milling by placing signal-carrying wires on top metal layers [22, 24–28]. The design is called an active shield because the wires on top metal layers are constantly monitored in order to detect an attack.

Figure 5.5 is one example constructed for the sake of illustration. The detection system generates a digital pattern with a pattern generator, sends them through mesh wires on top metal layer, and then compares received signals with copies from lower

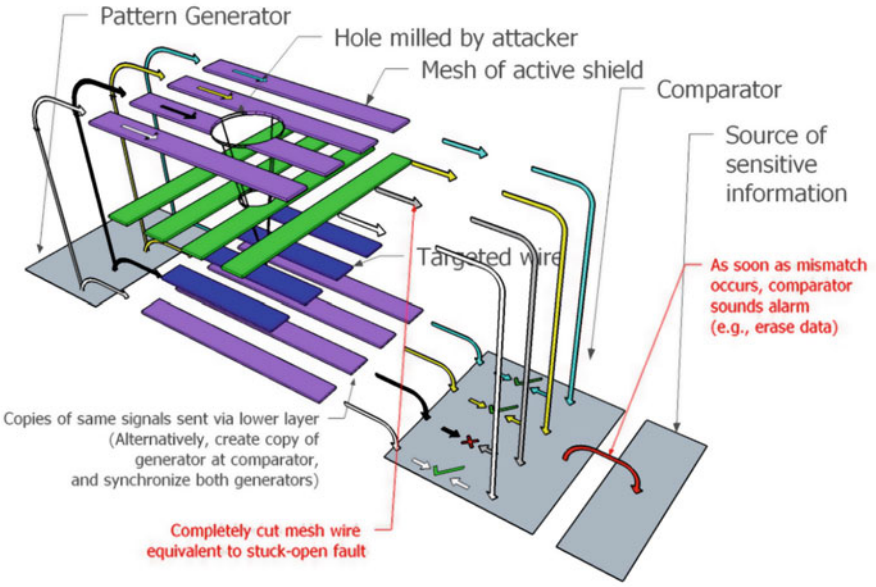

Fig. 5.5 Example of active shield for the purpose of milling detection

layer. If an attacker mills through the top layer mesh wires to reach targeted wire he will cut open some of them and cause a mismatch at the comparator. This will trigger the alarm, for example, stopping the generation of or destroying sensitive information.

Active shield as shown in Fig. 5.5 is known as a *digital* active shield, since it uses comparisons between digital signals to detect milling activity. *Analog* active shields also exist: for example, the authors in [24] use capacitance measurement on top layer shield mesh wires to detect damage done to it, and thereby detect tampering. A similar design [29] utilizes conductivity change in dielectric material due to exposure to ion irradiation, an essential step in FIB milling. Another design [25] uses a delay chain as a reference to compare against shield wire RC delay, and issues an alarm when mismatches between them are discovered. The problem with analog shield designs is that analog sensors rely on parametric measurement, which has been shown to become more difficult due to a number of issues, such as elevated process variation with feature scaling, improved milling capabilities that cause less disturbance, etc. [3]. These issues contribute to a higher error rate and make detection unreliable, thereby making analog active shield a less preferable approach.

5.3.2 Techniques to Attack and Secure Active Shields

Despite its popularity, active shields are not without problems. There are two main weaknesses of this approach, namely:

1. Active shields require large routing overhead, usually at least one entire layer;
2. Active shields have to be placed on top metal layers, which might not always be the most suitable layer.

We provide detailed discussions on both issues in the following subsections.

5.3.2.1 Routing Overhead

For thorough protection, active shields have to completely occupy at least one metal routing layer, which can be quite costly to implement. This is thought to be necessary because as long as attackers can afford extra probes, it is possible for him to reconstruct the desired signal from other related signals if they are not protected as well [30]. This requirement does not go well with designs having tight cost margin, or designs with few routing layers, which is especially true for devices such as smartcard, which are often fabricated with technology of larger dimensions such as 350 or 600 nm [2]. In fact, ICs designed for security critical applications such as smartcards, microcontrollers in mobile devices, and security tokens [2, 3, 31] are among the most common victims to this kind of attack. Lacking a very wide cost margin or a lot of routing layers precludes these devices from a number of new shield

designs that often incur area or routing overhead. Meanwhile, these same devices are in greatest need of evaluation to have realistic protection standards. Indeed, all designs eventually become outdated as new attacks are discovered, and these legacy generations of devices need up-to-date evaluation as well.

5.3.2.2 Stuck on Top Metal Layer

The "entire layer" requirement also leads to another problem, which is active shields must be placed on top-most metal layers. Due to the "entire layer" requirement, functional routings cannot penetrate shield layer without leaving an opening on the shield, therefore all routing layers above a shield layer will be unavailable to the functional design as well. Placing the shield on top-most metal layers saves routing layers for the functional design, but might come at cost of shield performance. Since top layers might not be best layers for them. In fact, top routing layers are known to have much larger minimum wire widths [32], making them less protective than lower layers [33].

Due to the popularity of the active shield approach, techniques have also been developed by attackers to neutralize it. One particularly expedient attack is called the *bypass attack* (Fig. 5.6). This attack utilizes an FIB milling tool with high aspect ratio, which allows the attacker to mill a hole so thin it's not going to completely cut off any mesh wires, therefore evading detection. Bypass attack is often the preferred method by the attacker because it does not require additional reverse engineering or circuit editing to reach the targeted wires, which saves his time and cost. Because of its advantage in speed, bypass attack is an especially serious challenge that all active shields must properly address. Another method exploits the fact that active shield functions by checking signals as received at end of shield wires. If the correct signals can be replicated, then the shield could be fooled. This is called a *reroute attack*. There can be a couple of ways to do this. One simpler approach is made possible when routing pattern of the shield wires allows rerouting, i.e., creating a short-cut with FIB so that a section of shield wire is made redundant [2], as shown in Fig. 5.7. That section of shield wire can then be removed without impacting signals received by the active shield, effectively creating an opening for probing. Another possible scenario is that attacker might reverse engineer the pattern generator so that correct signals can be fed from off-chip. In this case, the whole wire becomes redundant.

New active shield designs have been proposed to prevent these exploits. The authors in [22] investigated the problem of an attacker predicting the correct signals 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, whose seed comes from memory and fed by software. 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 reroute attack [27].

Perhaps the most serious threat to active shields comes from the capability of FIB technology itself. The capability of depositing metal as well as removing them

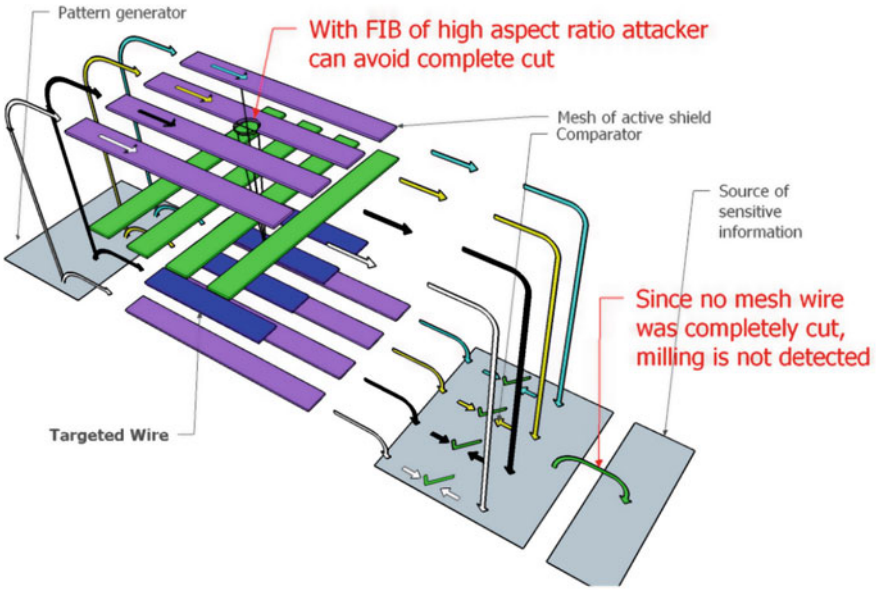

Fig. 5.6 Example of a bypass attack on active shield

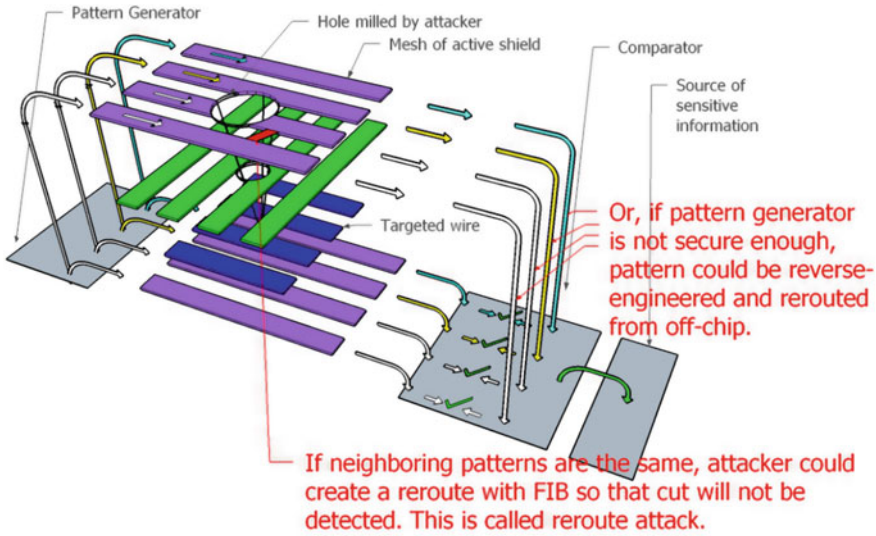

Fig. 5.7 Example of a reroute attack on active shield

effectively allows circuit editing, which enables the attacker to directly disable the active shield by editing its control circuitry or payload if it proves too difficult to bypass [3].

5.3.3 Other Antiprobing Designs

In addition to active shield designs, other approaches also exist. Authors in [34] presented a design to detect acts 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. However, it can only protect a certain number of specified nets from probing attacks using metal probes. Additionally, protection on certain specified nets could only be circumvented by probing at related signals instead [30]. Cryptographical techniques have also been proposed to address the probing attack. One cryptographical method called *t*-private circuits [35] 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. This method is cryptographically sound and does not require detection to deter probing, which eliminates the problem of protection design itself being disabled by the attacker. The proposed solution is also helpful in defeating side-channel attacks (SCA) and inspired further research in that area [36]. The weakness of this approach is it does incur quite large area overhead, and reliance on secure random number generation opens up questions on what happens when the random number generator is compromised, for example, with a probing attack. It has also been shown that such an approach might be jeopardized during CAD optimization process [37].

Some other approaches are often seen in security products in the market but less mentioned in academic literature. These approaches receive more discussion from researchers who publish successful attacks against security hardware in the market. For example, many security devices in the field also perform encryption of memories, for which a common problem is insufficient security when storing the keys for the encryption [31]. Some use one-time programmable (OTP) memories which can be read optically, or Electrically Erasable Programmable Read-Only Memories (EEPROM) which can be reprogrammed with ultra-violet (UV) light [38]. Other examples include scrambling of bus wires in the layout as a way to throw off attackers, password protection for boot strap loaders, and using one single sense signal for serpentine wires that elbow and bend to cover the layout [31]. These designs might not be as effective as they intended—since our knowledge of them is from their failure—but could serve as examples of "Do Nots" in antiprobing designs.

Another category consists of single-issue remedies that seek to disrupt specific steps during probing attacks. Many such techniques exist as recommendations by researchers who publish successful attacks [2, 3, 31]. These recommendations include methods to disrupt positioning of FIB for milling, avoid reusing hardware IPs to prevent attackers from copying existing reverse engineering knowledge against compromised IPs, methods to make IC packages more resistant to decapsulation, etc. Although not solutions on their own, these recommendations provide important insights into principles from the other side of the problem.

Table 5.1 Performance against known probing techniques of published designs

Designs	Protection against					
	Bypass shield	Rerouting attack	Disable shield	Back-side attack	Prediction attack	Related signals
Analog shield	Weak [3]	No	No		N/A	Yes
Random active shield [27]	Yes	Yes	No		No	Yes
Cryptographically secure shield [22]	Yes	Yes	No		Yes	Yes
PAD [34]	N/A	N/A	No	Some[a]	N/A	No

[a]PAD works for back-side attacks that require electrical contact to the targeted wires. It does not prevent passive back-side attacks such as PE

5.3.4 Summary on Antiprobing Protections

Based on known probing techniques (as was shown in Fig. 5.3) we may assess the protection of a few published designs, as shown in Table 5.1.

5.4 Layout-Based Evaluation Framework

In this section, we provide a motivation to investigate methods to evaluate antiprobing designs. We then discuss principles of antiprobing designs and rules to assess them, based on reviewing *modus operandi* of probing attacks. A layout-driven algorithm based on mainstream layout-editors is then introduced, which provides a quantitative evaluation of designs' vulnerabilities to probing attacks by reviewing their layout.

5.4.1 Motivation

Existing research on active shield designs focuses on preventing exploits. However, major problems exist with this approach. It negates the need of mass-produced and legacy generation security products of having realistic protection evaluations, and gives us a false sense of security despite the fact that security is always relative. Should designers focus on beating latest exploit into current shield designs, inherent problems like these tend to get overlooked. Further, circuit editing capability of the FIB actually makes it impossible for active shields to have absolute security: attacker can always choose to edit out the shield itself. Despite that, having a design

that beats all known exploits can make people dangerously comfortable with a false sense of security and forget that fact. This reality has put the ability to evaluate a design for vulnerability to probing attacks in dire need.

Evaluation contributes to existing antiprobing design flow in a few ways (Fig. 5.8):

1. First, the evaluation tool can be used to create a feedback for the design process, and help designers in pursuit of an optimized design (Fig. 5.8a).
2. In addition, comparisons can be made between security evaluations of certain representative designs. By carefully controlling design methods and parameters, design principles such as trade-offs in antiprobing designs can be identified and investigated (Fig. 5.8b).
3. In addition to identifying design principles, evaluation also enables us to study new design ideas, and optimize them into new design strategies, or establish a solid understanding of why they wouldn't work (Fig. 5.8c).

5.4.2 Assessment Rules

Before presenting the framework to assess protection designs against probing attacks, it is essential to establish the principles of antiprobing 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 probing attack, it is important to remember that attackers capable of nanometer scale milling are not restricted to probing 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 [3]. As a result, while designs that can defeat all known attacks might not be impossible, it is certainly impractical to pursue for most devices.

Meanwhile, another myth is to underestimate the difficulty of probing 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 probing 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

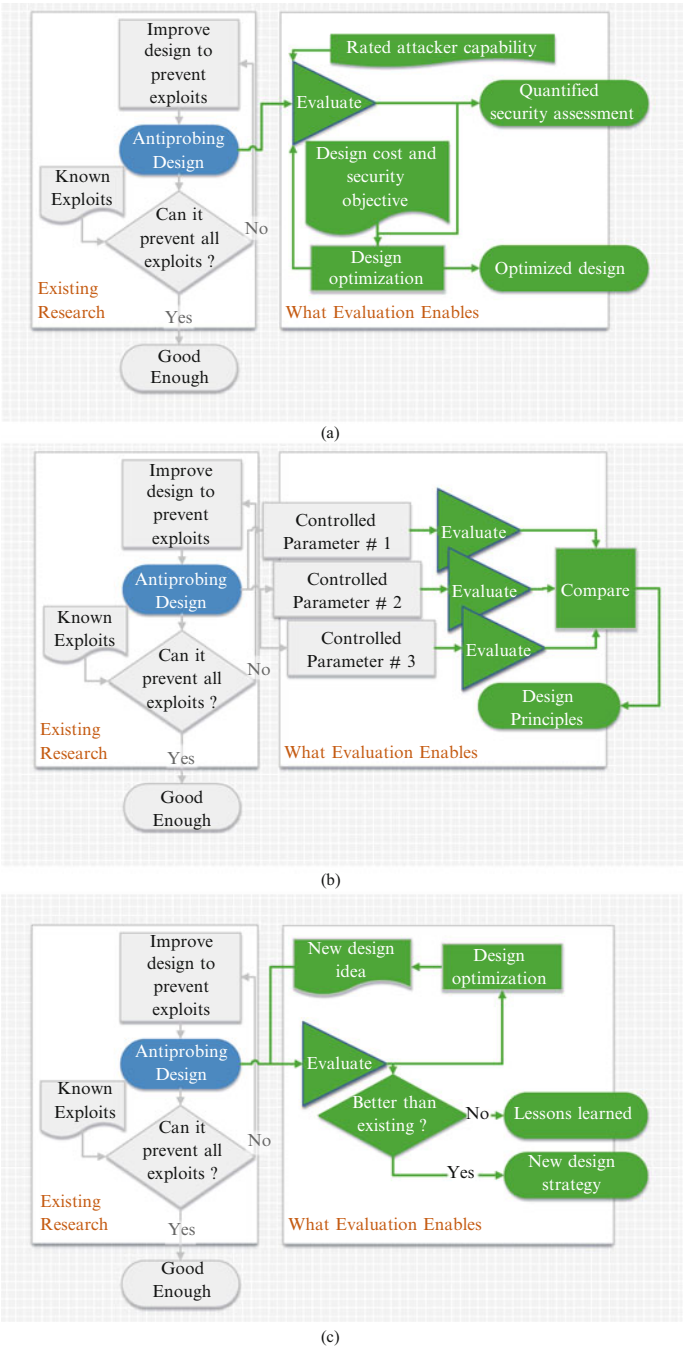

Fig. 5.8 Applications of evaluation in antiprobing design

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.

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

- For each necessary step during a probing 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;
- Represent the protection against attackers with infinite resources by the sum of techniques with the lowest time cost from each necessary step;
- For protection against less well-equipped attackers, repeat the same process without techniques requiring unavailable capabilities.

Under these rules it is possible for a particular probing 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, infinite time cost is unlikely to appear for an entire step 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* [3].

From the assessments of existing antiprobing designs 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* of targeted wires.

5.4.3 State-of-the-Art Active Shield Model

To accurately evaluate exposed area of targeted wires in a layout, a mathematical model of detection mechanism is necessary. And to choose a proper mathematical model, one must first establish a set of reasonable assumptions about the detection system. The complete mathematical equations have been covered in [33]; in this section we place the emphasis on establishing a set of assumptions that define a reasonably well-designed active shield to the best of our knowledge based on published research in the area.

In this study, we assume attacker performs front-side milling with FIB technology as shown in Fig. 5.4a. The hollowed-out cone shown in the figure represents a hole milled with FIB equipment. In reality, a milled hole for the purpose of probing attack will probably be larger than the hollowed-out cone, since the probes need to maintain a reliable connection. Here we consider the cone shown in Fig. 5.4a as a best-case scenario for the attacker and worst-case scenario for the designer, so that a margin of safety can be left in favor of the shield.

Active shield designers are interested in the scenario where the attacker would make a mistake and completely cut off (at least) one shield wire. This *complete cut* event is desirable because it will make detection easy and robust. It is possible that a *partially cut* event may be detected in ways similar to the analog shield idea [24]; however, designers usually only consider complete cuts, and with good reason.

From an electrical engineering point of view, a partially cut wire creates a point of high resistivity and current density on the wire. For the timescale relevant to the probing attack and its detection, this manifests mainly as increased delay on the timing arc of the shield wire. On-chip timing measurement can be done with a frequency counter, a vernier delay chain, a current ramp and an analog-to-digital (A/D) converter, or a time-to-voltage converter [39]. None of them is known for being low on area overhead, or even close to the area overhead of a simple XOR gate per wire. Further, consider the requirement for the active shield to cover the entire functional layout, or at least all wires that might leak sensitive information if probed, the area overhead problem is astronomically worsened. To keep the area overhead manageable, such a measurement have to be handled either by a shared measurement module switched among all shield wires, or by only monitoring a few targeted wires, as is done in Probe Attempt Detector (PAD) [34]. The disadvantage of the PAD approach is discussed in Sect. 5.3.3; For the switched solution, each wire has to be compared against its own normal state and incur astronomical area overhead on memories, or use a constant reference and risk false alarms and/or escapes due to environmental and fabrication variation. Margin will have to be left to accommodate false alarms, then the same margin could also be used by an attacker to slip in undetected. In either case, much is traded for possible improvement of slightly higher FIB aspect ratio the shield is probably able to detect, depending on environmental variations.

In summary, partial cut detection greatly complicates the problem for dubious gains. The few (one, in fact) attempts that tries to do this to our knowledge [25] are rather unconvincing as the proposal(s) can be quite fittingly covered with the "constant reference" option of our hypothetical "switched" solution, did not investigate the quite obvious weakness of false alarms and/or escapes, only performed simulation that verified the intended function, and committed other design flaws such as creating elbows and bends in shield wires that allows reroute attacks. So far, we have yet to see a sound shield design that can claim this feat. Therefore, in this study we focus on detection method based on complete cuts.

As was discussed previously in Sect. 5.3.2, a reroute attack 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 [2]. This forces active shield designs to only use parallel wires of minimum spacing and widths [22], without bending or elbows. In this case, the best placement of center of the milling (i.e., least likely to result in a complete cut of a wire) by the attacker is to place it at the middle of the space between any two wires. Conversely, the designer need to ensure within a certain radius d_{eff} of the center of the milling, the milled hole is at least deep enough to cut open the two closest shield wires. If we further assume an aspect ratio of the FIB R_{FIB}, then these two conditions together create a

restriction of milling hole diameter d. Since aspect ratio of the FIB R_{FIB} is a ratio between milling hole diameter d and depth D (which, if we assume the designer knows where his possible targeted wires are, is a known variable), this requirement translates into a maximum aspect ratio of the FIB $R_{\text{FIB, max}}$ the hypothetical active shield can detect, provided the milling is performed perpendicular to the IC.

5.4.4 *Impact of Milling Angle upon Effect of Bypass Attack*

One interesting question here is whether attacker would benefit if instead of milling vertically, he mills at an angle, as shown in Fig. 5.9. This question is directly relevant to appraising the threat of bypass attack: If the answer to the question is no, that means conventional vertical milling is the best-case scenario for the attacker in terms of possibility to bypass the shield; in that case, the security of any given active shield design against bypass attack can be simply deduced with the aspect ratio of the FIB milling tool it is able to detect. Otherwise, the problem of protecting against bypass attack would become much more complicated. If we assume the attacker were able to mill at an angle $\theta \leq \frac{1}{2}\pi$ relative to the surface of the IC, then he would cut off wires within region d'_{eff} instead of d_{eff}. We may evaluate d'_{eff}, and then taking derivative of $\frac{d'_{\text{eff}}}{d_{\text{eff}}}$ and letting it equal to zero yields the minimum d'_{eff} and angles it is reached. If we further assume shield wire height/width ratio $(A/R) = 2.5$ as in [32] (ITRS uses 2.34 [9]), minimum d'_{eff} over d_{eff} yields results shown in Table 5.2 for typical FIB aspect ratio R_{FIB}. 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 bypass

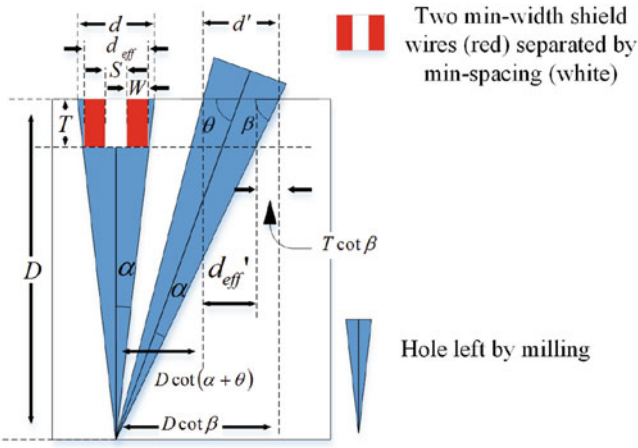

Fig. 5.9 Geometric calculations for non-perpendicular milling scenario

Table 5.2 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

attack is considered a convenient and preferable approach [3], this suggests that it has the potential of becoming much more devastating. This result shows us that bypass attack is not a simple one dimensional issue, because obviously the benefit of milling at an optimized angle would be diminished if that angle would not agree with the aforementioned "milling center at middle between two wires" rule. Whether this reduction in milling diameter can be achieved will depend on the relative position of the shield wires with regard to targeted wires. In other words, this potential threat must be addressed by optimized positioning of the targeted wires, so that attacker cannot further reduce the possibility of complete cut by cleverly mill at an angle. This will require optimization based on layout information, which is best handled by a tool integrated with layout editors.

5.4.5 Algorithm to Find Exposed Area

We solve the problem of finding exposed area on targeted wires by first finding the complement area, i.e., the area that attacker wouldn't want to mill in. Consider wires above targeted wires in the layout. If the attacker completely cuts off a shield wire, he risks detection; even if it is merely a functional wire, he still risks corrupting information he wants to access without complete knowledge of its function through extensive reverse engineering. Using mathematical models shown in [33], complete cut will happen if center of milling exists within d_{faredge} from the far edge of the wire, where

$$d_{\text{faredge}} = \frac{D - H}{2R_{\text{FIB}}}$$

$$d_{\text{faredge}} = \frac{(2W + S)D}{2(2W + S)R_{\text{FIB}} + (A/R)W} \tag{5.1}$$

where d_{faredge} is the maximum distance from the center of the hole to the far side of the wire (shown in Fig. 5.10), where d is the diameter of the hole, W and S are minimum width and spacing of the shield layer, D is the depth of the hole, (A/R) is the aspect ratio of the shield wire metal, H is the thickness of the intersecting wire, and R_{FIB} 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. Also, note that d_{faredge} represents the radius within which the milling will cut deep enough to completely cut open wires; therefore d_{faredge} is shorter than the radius ($\frac{1}{2}d$) of the hole radius left on the shield layer (as shown in Fig. 5.10a, b).

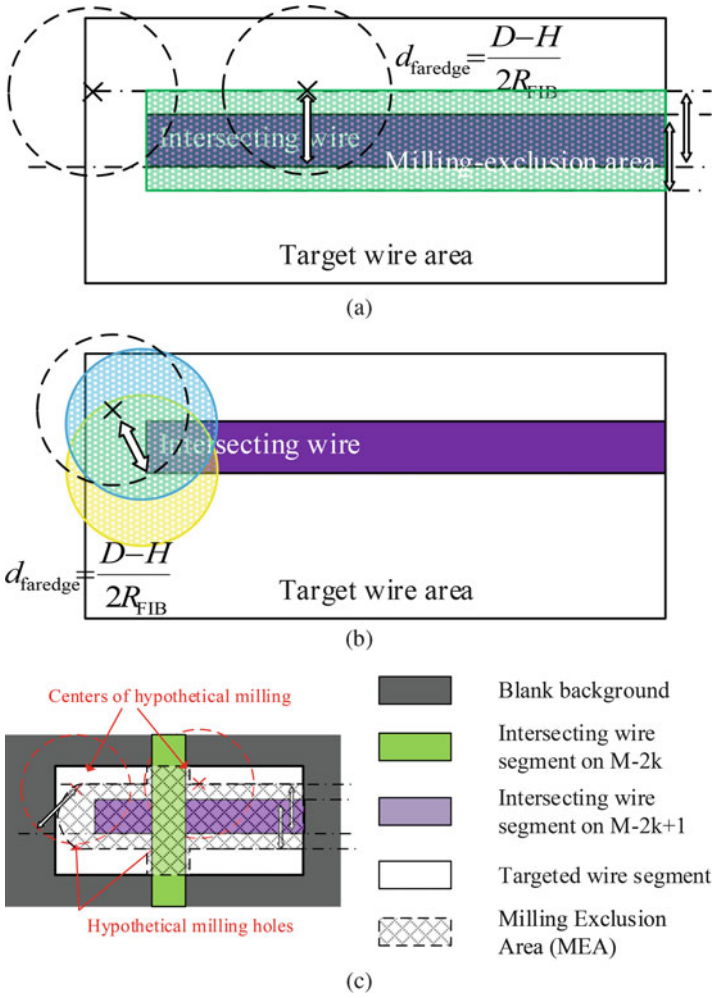

Fig. 5.10 Finding milling-exclusion area. (**a**) Milling-exclusion area on sides of intersecting wire; (**b**) Milling-exclusion area on ends of intersecting wire; (**c**) Complete milling-exclusion area in the presence of multiple intersecting wires

Equation 5.1 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 5.10 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 Fig. 5.10a, 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 Fig. 5.10b. 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 milling center locations that will guarantee cut off both corners, i.e., a complete cut. Consequently, any rectangular intersecting wire will project a milling-exclusion area whose shape is the union of the shape shown in Fig. 5.10a, b.

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. Layout design tools, such as Synopsys IC Compiler, are able to provide sufficient information to determine each of these constituent rectangular wires in the form of their corner coordinates, with which a bit-map of mill-exclusion areas can be easily produced. 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 probing attack. This process is elaborated in the pseudocode as shown in Algorithm 5.1.

Algorithm 5.1: Proposed locator algorithm for exposed area.

```
   Input: targeted_nets, precision, all_layers
   Output: draw.script
1  begin
2      targeted_wire_shapes ⇐ get_net_shapes(targeted_nets)
3      N ⇐ sizeof_collection(targeted_wire_shapes)
4      for (i = 1:N) do
5          targeted_wire_shape ⇐ targeted_wire_shapes(i)
6          canvas_size ⇐ get_sizes(get_bounding_box(targeted_wire_shape))*precision
7          Print command in draw.script to create canvas in draw.script whose size equals to canvas_size
8          layers_above ⇐ get_layers_above(all_layers, get_layerof(targeted_wire_shape))
9          M ⇐ sizeof_collection(layers_above)
10         for (j = 1:M) do
11             this_layer ⇐ layers_above(j)
12             d_faredge_on_thislayer ⇐ (D−H)/(2R_FIB)
13             intersecting_wire_shapes ⇐ get_net_shapes(targeted_nets) in
               get_bounding_box(targeted_wire_shape) on this_layer
14             L ⇐ sizeof_collection(intersecting_wire_shapes)
15             for (k = 1:L) do
16                 intersecting_wire_shape ⇐ intersecting_wire_shapes(k)
17                 Print command in draw.script to create projection in draw.script whose radius/widths
                   equals to d_faredge_on_thislayer
18             end
19         end
20     end
21 end
```

As shown in Algorithm 5.1, the presented methodology starts with a set of logic nets. The algorithm first identifies their constituting wire shapes in

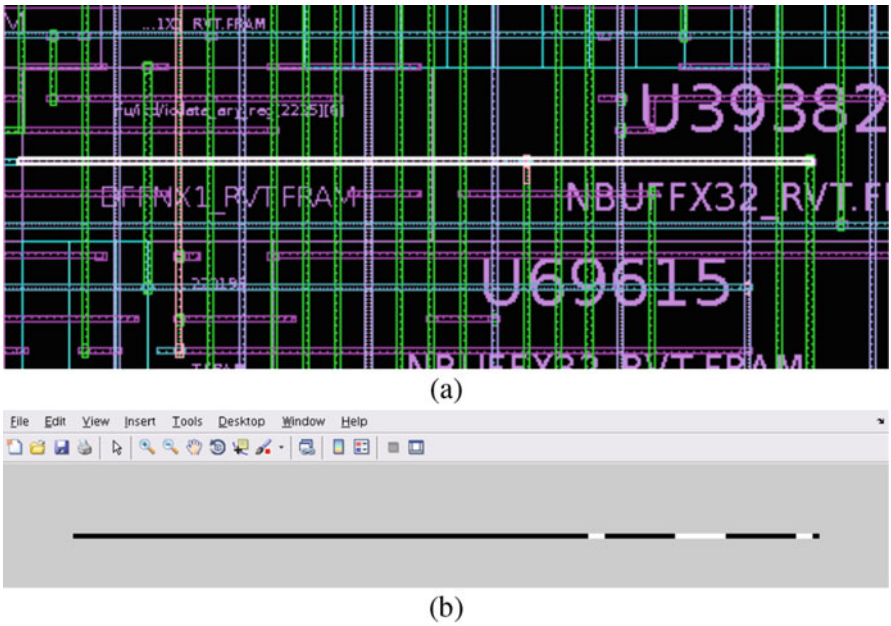

Fig. 5.11 Exemplary results produced by proposed algorithm. (**a**) Exemplary targeted wire (*highlighted*) in layout; (**b**) Mill-exclusion area (*black*) projected on canvas of same wire

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 Fig. 5.11). 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 probing 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 presented algorithm rounds towards minus infinity on borders, i.e., errors 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.

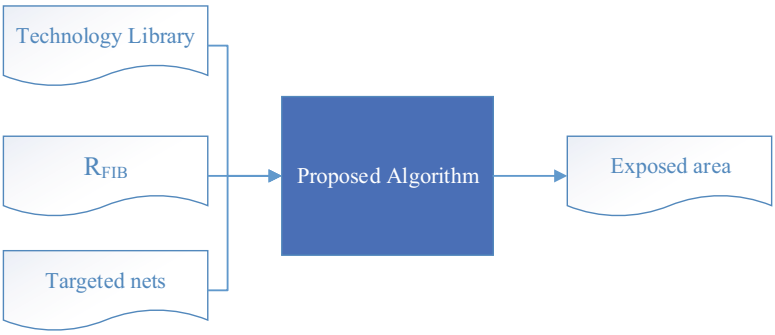

Fig. 5.12 A simplified diagram of presented algorithm to find exposed area

5.4.6 Discussions on Applications of Exposed Area Algorithm

Figure 5.12 shows a simplified diagram of Algorithm 5.1, where only inputs and output of the algorithm are shown. One apparent observation is that by controlling inputs to the algorithm, impacts of these inputs upon exposed area of targeted wires can be studied: for example, controlling R_{FIB} lets us study the threat of improved attacker sophistication, controlling targeted nets allows us to study benefits by relocating these wires, etc.

One important question is how the user should choose targeted nets to prepare against a serious attack. To start, it can be assumed that a designer would have a rough idea of which nets might attract most attention; and according to [30], linear combinations of these nets should also be considered. However, although it is always better safe than sorry, it might not be realistic to consider all possible candidates in this list, as it can be too many to process. Luckily, locating targeted wires remains a problem in published attacks [3]; and since we do not consider back-side attacks in this study, we can safely assume a minimum positioning error on the attacker, and rule out those nets that do not have a large enough area for attacker to reliably locate. Such an assumption along with assumption on attacker's R_{FIB} would constitute the quantified attacker capability model we advocated in Fig. 5.8a.

5.5 Conclusion

Probing attacks is defined as direct access of sensitive information by probing the physical wires that carries it. Reaching such wires necessitates milling, which is often satisfied by employing focused ion beam milling. Active shields detect milling by covering the layout with signal-carrying wires whose signals are monitored. Despite recent advances such as back-side attacks, their own limitations and lack of a better option have kept active shield as the most popular approach to

secure an IC against probing attacks. However, the active shield design is plagued with weaknesses that could be exploited by attackers. So far, existing research concentrated on securing the design against these weaknesses, at the cost of incurring high hardware cost and making themselves prohibitive to common victims to probing attacks such as smartcards and security tokens. In this chapter, we have reviewed existing probing attacks and antiprobing research, and presented a layout-driven framework to assess designs for vulnerabilities to probing attacks. Based on design principles and assessment rules we have established considering reported successful probing attacks, we presented an algorithm to analyze layout designs for potential vulnerabilities to probing attacks. We expect work presented here to serve as a basis for future methodologies to protect against probing attacks that are more effective, require lower hardware cost and applicable to a wider variety of ICs.

References

1. S. Skorobogatov, Physical attacks on tamper resistance: progress and lessons, in *Proceedings of 2nd ARO Special Workshop on Hardware Assurance*, Washington (2011)
2. C. Tarnovsky, Tarnovsky deconstruct processor, Youtube (2013) [Online]. Available: https://www.youtube.com/watch?v=w7PT0nrK2BE
3. V. Ray, Freud applications of fib: invasive fib attacks and countermeasures in hardware security devices, in *East-Coast Focused Ion Beam User Group Meeting* (2009)
4. R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems* (Wiley, New York, 2001)
5. WIRED, How to reverse-engineer a satellite tv smart card (2008) [Online]. Available: https://youtu.be/tnY7UVyaFiQ
6. K. Zetter, From the eye of a legal storm, murdoch's satellite-tv hacker tells all (2008) [Online]. Available: http://www.wired.com/2008/05/tarnovsky/
7. Invasive attacks (2014) [Online]. Available: https://www.sec.ei.tum.de/en/research/invasive-attacks/
8. I. Huber, F. Arthur, J.M. Scott, The role and nature of anti-tamper techniques in us defense acquisition, DTIC Document, Tech. Rep. (1999)
9. International Technology Roadmap for Semiconductors, 2013 edn., Interconnect (2013). [Online]. Available: http://www.itrs2.net/2013-itrs.html
10. X. Zhuang, T. Zhang, H.-H.S. Lee, S. Pande, Hardware assisted control flow obfuscation for embedded processors, in *Proceedings of the 2004 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*. Ser. CASES '04 (ACM, New York, 2004), pp. 292–302. [Online]. Available: http://doi.acm.org/10.1145/1023833.1023873
11. R.S. Chakraborty, S. Bhunia, Harpoon: an obfuscation-based soc design methodology for hardware protection. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **28**(10), 1493–1502 (2009)
12. S.E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, M. Tehranipoor, A survey on chip to system reverse engineering. ACM J. Emerg. Technol. Comput. Syst. **13**(1), 6 (2016)
13. V. Sidorkin, E. van Veldhoven, E. van der Drift, P. Alkemade, H. Salemink, D. Maas, Sub-10-nm nanolithography with a scanning helium beam. J. Vac. Sci. Technol. B **27**(4), L18–L20 (2009)

14. Y. Fu, K.A.B. Ngoi, Investigation of aspect ratio of hole drilling from micro to nanoscale via focused ion beam fine milling, *Proceedings of The 5th Singapore-MIT Alliance Annual Symposium*. http://web.mit.edu/sma/about/overview/annualreports/AR-2004-2005/research/research06imst10.html
15. H. Wu, D. Ferranti, L. Stern, Precise nanofabrication with multiple ion beams for advanced circuit edit. Microelectron. Reliab. **54**(9), 1779–1784 (2014)
16. H. Wu, 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. J. Mater. Sci. Mater. Electron. **25**(2), 587–595 (2014)
17. C. Helfmeier, D. Nedospasov, C. Tarnovsky, J.S. Krissler, C. Boit, J.-P. Seifert, Breaking and entering through the silicon, in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (ACM, New York, 2013), pp. 733–744
18. C. Boit, C. Helfmeier, U. Kerst, Security risks posed by modern ic debug and diagnosis tools, in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)* (IEEE, Washington, 2013), pp. 3–11
19. A. Schlösser, D. Nedospasov, J. Kramer, S. Orlic, J.-P. Seifert, Simple photonic emission analysis of aes, in *Cryptographic hardware and embedded systems–CHES 2012* (Springer, Heidelberg, 2012), pp. 41–57
20. C. Boit, Fundamentals of photon emission (PEM) in silicon - electroluminescence for analysis of electronics circuit and device functionality, in *Microelectronics Failure Analysis* (ASM International, New York, 2004), pp. 356–368
21. C. Boit, R. Schlangen, U. Kerst, T. Lundquist, Physical techniques for chip-backside ic debug in nanotechnologies. IEEE Des. Test Comput. **3**, 250–257 (2008)
22. J.-M. Cioranesco, J.-L. Danger, T. Graba, S. Guilley, Y. Mathieu, D. Naccache, X.T. Ngo, Cryptographically secure shields, in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (IEEE, Arlington, 2014), pp. 25–31
23. Y. Xie, C. Bao, C. Serafy, T. Lu, A. Srivastava, M. Tehranipoor, Security and vulnerability implications of 3D ICs. IEEE Trans. Multiscale Comput. Syst. **2**(2), 108–122 (2016)
24. P. Laackmann, H. Taddiken, Apparatus for protecting an integrated circuit formed in a substrate and method for protecting the circuit against reverse engineering, 28 September 2004, US Patent 6,798,234
25. M. Ling, L. Wu, X. Li, X. Zhang, J. Hou, Y. Wang, Design of monitor and protect circuits against fib attack on chip security, in *2012 Eighth International Conference on Computational Intelligence and Security (CIS)* (IEEE, Guangzhou, 2012), pp. 530–533
26. A. Beit-Grogger, J. Riegebauer, Integrated circuit having an active shield, 8 November 2005, US Patent 6,962,294. [Online]. Available: https://www.google.com/patents/US6962294
27. S. Briais, J.-M. Cioranesco, J.-L. Danger, S. Guilley, D. Naccache, T. Porteboeuf, Random active shield, in *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)* (IEEE, Leuven, 2012), pp. 103–113
28. Invia., Active Shield IP (digital IP protecting System-on-Chip (SoC) against tampering through a metal mesh sensor) (2016) [Online]. Available: http://invia.fr/detectors/active-shield.aspx
29. F. Ungar, G. Schmid, Semiconductor chip with fib protection, 2 May 2006, US Patent 7,038,307. [Online]. Available: https://www.google.com/patents/US7038307
30. L. Wei, J. Zhang, F. Yuan, Y. Liu, J. Fan, Q. Xu, Vulnerability analysis for crypto devices against probing attack, in *2015 20th Asia and South Pacific Design Automation Conference (ASP-DAC)* (IEEE, Tokyo, 2015), pp. 827–832
31. C. Tarnovsky, Security failures in secure devices, in *Black Hat Briefings* (2008)
32. Freepdk45: Metal layers (2007) [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45:Metal_Layers
33. Q. Shi, N. Asadizanjani, D. Forte, M.M. Tehranipoor, A layout-driven framework to assess vulnerability of ics to microprobing attacks, in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (2016)

34. S. Manich, M.S. Wamser, G. Sigl, Detection of probing attempts in secure ICs, in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (IEEE, San Francisco, 2012), pp. 134–139
35. Y. Ishai, A. Sahai, D. Wagner, Private circuits: securing hardware against probing attacks, in *Advances in Cryptology-CRYPTO 2003* (Springer, Heidelberg, 2003), pp. 463–481
36. M. Rivain, E. Prouff, Provably secure higher-order masking of aes, in *Cryptographic Hardware and Embedded Systems, CHES 2010* (Springer, Heidelberg, 2010), pp. 413–427
37. D.B. Roy, S. Bhasin, S. Guilley, J.-L. Danger, D. Mukhopadhyay, From theory to practice of private circuit: a cautionary note, in *2015 33rd IEEE International Conference on Computer Design (ICCD)* (IEEE, Washington, 2015), pp. 296–303
38. D. T. Ltd., Known attacks against smartcards (2015) [Online]. Available: http://www.infosecwriters.com/text_resources/pdf/Known_Attacks_Against_Smartcards.pdf
39. T. Xia, On-chip timing measurement. Ph.D. dissertation, University of Rhode Island (2003)