A Physical Design Flow against Front-side Probing Attacks by Internal Shielding*

Huanyu Wang, Qihang Shi, Navid Asadizanjani, Domenic Forte, and Mark M. Tehranipoor
ECE Department, University of Florida
{huanyuwang, qihang.shi}@ufl.edu, {nasadi, dforte, tehranipoor}@ece.ufl.edu

ABSTRACT

Security-critical applications on integrated circuits (ICs) are threatened by microprobing attacks that extract sensitive information through focused ion beam (FIB) based milling. Existing countermeasures, such as active shield, analog shield and t-private circuit, have proven to be inefficient and provide limited resistance. In this paper, we propose a FIB-aware anti-probing physical design flow to reduce the vulnerability of security-critical nets in a design. Results show that our proposed technique can reduce the vulnerable exposed area on critical nets to probing attack by 90% in AES and DES modules with only 5% area overhead.

1. INTRODUCTION

With the rapid development of integrated circuits (ICs) and increasing reliance on electronic systems, the risk of leaking security-critical information, such as keys, firmware, device configuration and protected data, stored in ICs through software and hardware based attacks is higher than ever before. Although countermeasures against software and non-invasive hardware attacks, e.g., side channel and fault injection attack have been widely investigated, there is no efficient protection method against physical probing attacks. In a probing attack [2], the internal wires of security-critical applications, such as smartcards, smartphones, military systems, and financial systems, are physically contacted to extract sensitive information, e.g. encryption keys or confidential data. With the help of focused ion beam (FIB) [1], a powerful circuit editing tool that can mill and deposit material with nanometer level precision, an attacker can bypass protection mechanisms and reach wires carrying sensitive information [3, 4]. Note that FIB's resolution is keeping pace with technology scaling and attacker does not need to purchase a new one since rent it by time or buy a used one is quite low cost. In the Internet-of-Things (IoT) era, the threat from probing is aggravated since there will be a larger volume of low-end devices which are physically accessible.

The main issues surrounding common countermeasures against probing attacks are their prohibitive area, timing and power overhead as well as their ad hoc nature [9]. Currently, there is no holistic and efficient approach that can be easily incorporated into conventional application-specific integrated circuit (ASIC) design flow to protect security-critical circuits and nets from FIB-based probing. In this paper, our major contributions are summarized as follows:

- A physical design flow that is straightforward to incorporate into conventional ASIC design flow and mitigates the threat of front-side probing attacks.
- An internal shielding mechanism that is implemented by automatic place-and-route of existing electronic design automation (EDA) tools without inserting extra shielding and pattern generator circuit, which avoids large area overhead. The shield nets are selected from existing design using a new metric.
- A metric is developed to identify security-critical nets which are most likely to be targeted for probing attacks. Such nets include those directly connected to the security asset as well as nets in the asset's fanout from which sensitive information could be derived.
- A method is developed to choose the best shield layer which can provide the optimal protection to target nets based on the technology specifications.
- Our proposed approach is evaluated on AES and DES modules. Results show that the area vulnerable to probing attacks decreases by 90% with only 5% area overhead on AES/DES.

The rest of the paper is organized as follows. In Section 2, we provide our threat model of probing attacks. In Section 3, we present our probing-aware design flow including target/shield identification, shield layer selection, constrained layout, and exposed area calculation. The evaluation results are provided in Section 4. Finally, we conclude in Section 5.

2. THREAT MODEL

In this paper, we restrict our focus on electrical probing from the front-side. Back-side and optical probing will be addressed in future work. The objective of the adversaries is to extract assets stored in a device through probing attack. We further assume a strong attacker that has full layout information of the design from either reverse engineering or a rogue employee in the foundry. We presume the attack is performed by milling a hole using FIB technology and probing at the sensitive net exposed by the milled hole. We also assume that attack detection is conservative, i.e., requiring a complete cut of shield wires, due to the unreliability of detecting partial cuts.

3. ANTI-PROBING DESIGN FLOW

*This work was supported in part by Semiconductor Research Corporation (SRC).

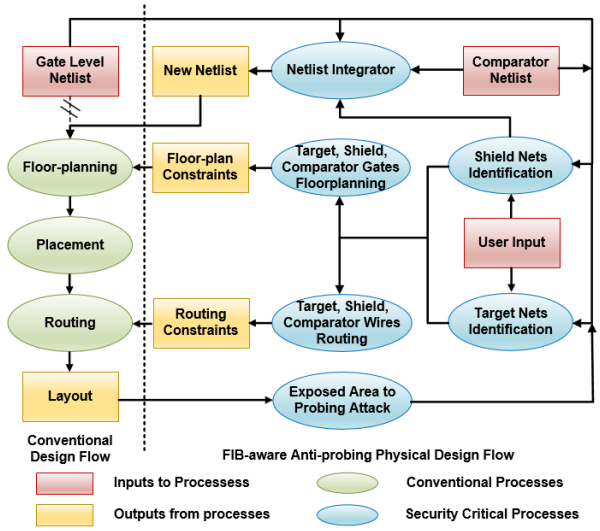

Fig. 1: Overall FIB-aware anti-probing physical design flow.

Our objective is to develop a FIB-aware anti-probing physical design flow that incorporates automated security-aware floor-planning, cell placement, and routing steps into the conventional flow in order to protect the security-critical nets against front-side probing attacks (note protection from back-side is outside the scope of this paper). We shall accomplish this by using a chip's internal functional nets as shield nets on upper layer to provide coverage for "target" nets on lower layers (i.e., those carrying assets) in the design. Another copy of shield nets will be routed in lower layer. Once one of the shield nets on upper layer is cut off in an attack, a *comparator* will detect the mismatch of the signal on upper shield net and the one from lower copy. Then an alarm will be triggered to the CPU or micro-controller to take the appropriate actions (e.g., terminate the operation of the chip or remove all asset information). The overall workflow of our anti-probing physical design flow is shown in Fig. 1.

3.1 Target Net Identification

In this Section, we discuss how we identify the nets which are most likely to be targeted for probing. Nets that are connected to assets are the most likely to be targeted by an attacker. In addition, an attacker can also target nets that are not directly connected to an asset, but still contain information from which the asset can be derived. Therefore, in addition to nets that are directly connected to asset nets, other nets which can be exploited to extract the asset also need to be protected against probing attack. Since it's inefficient and impossible to protect all nets in a SoC, we develop a probing target identification metric to rank the nets according to their ability to leak asset information and therefore, the nets' likelihood of being targeted for probing can be deduced.

Our anti-probing design flow first requires the designers to input the name of nets/ports where the asset is located, e.g., the name of key nets. Then our flow performs the target net identification technique to identify all nets which are likely to be targeted for probing attack. This technique utilizes a *Target Score* ($f_{\mathrm{TS}}(i)$) metric to identify the target nets. For each net i in the circuit:

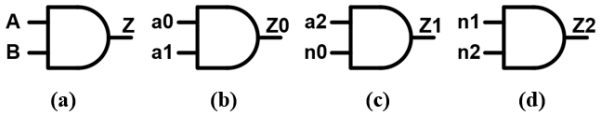

Fig. 2: AND gate examples.

Table 1: Target score calculation for nets in Fig. 2.

Measures	a0	a1	Z0	a2	n0	Z1	n1	n2	Z2
$f_{\mathrm{IL}}(i)$	1	1	1	1	0	1	0	0	0
$f_{\mathrm{PD}}(i)$	0	0	0	0	1	1	0	0	0
$f_{\mathrm{TS}}(i)$	1	1	1	1	0	1/2	0	0	0

$$f_{\mathrm{TS}}(i) = \frac{f_{\mathrm{IL}}(i)}{f_{\mathrm{PD}}(i)+1} \tag{1}$$

where $f_{\mathrm{IL}}(i)$ denotes information leakage and quantifies the amount of asset information leaked by observing net i, and $f_{\mathrm{PD}}(i)$ indicates the difficulty in propagating an asset signal to net i. A larger value of $f_{\mathrm{IL}}(i)$ means more asset information can be leaked at net i. On the other hand, a larger $f_{\mathrm{PD}}(i)$ value indicates that it is more difficulty in propagating an asset signal to net i. Hence, a higher $f_{\mathrm{TS}}(i)$ represents a higher likelihood of being targeted for probing.

***Target Score* Calculation:** Table 1 shows the Target Score calculation in Figs. 2(b)(c)(d), assuming n0, n1, n2 are non-asset primary inputs. In Fig. 2(d), since both inputs are non-asset nets without any information leakage, the *Target Score* for Z2 is 0. Note that the asset should be identified by the chip designer as a user input of our anti-probing design flow as shown in Fig. 1. If one of the assets is not identified in the user input, the target net identification would be unable to recognize the nets that can leak sensitive information of the unidentified asset. In addition, for those nets that might be utilized to infer asset information through complicated mathematical analysis, e.g. the intermediate nets of an encryption/decryption process used in differential fault analysis (DFA) technique, they are not covered by the target net identification, so they should be declared in user input as a special "asset" to be protected against probing attack.

3.2 Shield Net Identification

One unique feature that distinguishes our proposed anti-probing physical design flow from previously proposed techniques, is the adoption of internal functional nets of the design as shield to protect target nets against probing attack. Existing active shield countermeasures are vulnerable to bypass attacks [11] and reroute attacks [12] because the shield at the top-most layer is relatively easy to access and manipulate. In addition, more advanced active shields require cryptographically secure pattern generators [5], which themselves are sources of vulnerability and additional overhead. In contrast, utilizing internal functional nets provides the following major advantages. First, they will be routed within internal layers of the chip and therefore far more difficult to bypass and reroute. Second, the design itself will generate these signals alleviating the need for pattern generation, which will save the major area overhead introduced by active shield pattern generation. In this design, we develop a

technique for identifying which internal nets can be utilized as shielding nets (covering nets). We define the following five requirements along with associated metrics as follows: Target score; Toggle frequency; Switching probability; Controllability; Delay slack.

For each of the aforementioned shield requirements, a threshold value of corresponding metric should be determined to maximize the coverage on target nets and minimize the vulnerabilities and impacts from shield nets. The final shield candidate nets will be the intersection of the five net collections which satisfy the threshold values for each shield requirement.

3.3 Best Shield Layer

After appropriate shield nets are identified, the metal layer in the chip layout to route these shielding nets needs to be determined. In this paper, we consider a milling scenario using FIB technology as shown in Fig. 3, where colored bars are used to represent metal wires on different routing layers. 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 shield layer (blue in Fig. 3).

We use shield security to represent the maximum FIB aspect ration that the shield can protect against. The higher of shield security value, the better of the shield. The shield security can vary depending on shield layer, target layer, width of shield wire, thickness of shield wire and other layout technology parameters. Therefore, different technology library might derive different shield security and different best shielding layer. Table 2 shows the shield security calculated from SAED32nm library. As we can see, when target nets are on layer 1, 2, and 3, layer 6 is the best shield layer. It is because for target nets on these layers, the shield security value for shield layer 6 is always the largest among all shield layers. Therefore, in our implementation, target nets are routed under metal 4 and shield nets are routed on metal 6.

3.4 Floor-planning Constraints

In conventional design flows, CAD tools perform floorplanning to optimize timing, power, and area. In an original design as shown in Fig. 4(a), target nets and the blocks containing them (red) are distributed randomly throughout the design. It is neither easy nor efficient to protect them

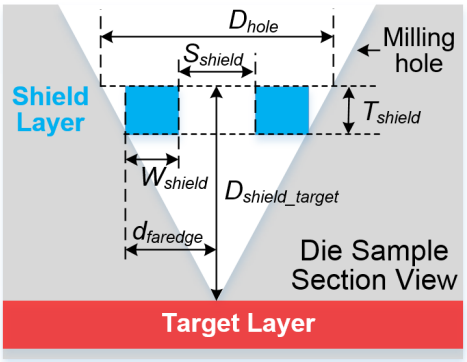

Fig. 3: Geometric calculations for perpendicular milling scenario.

Table 2: Shield Security in SAED32nm library.

Max R_{FIB}	Shield Layer							
Target Layer	9	8	7	6	5	4	3	2
8	0.19	N/A						
7	0.41	0.39	N/A					
6	0.62	0.84	0.39	N/A				
5	0.84	1.29	0.84	0.71	N/A			
4	1.06	1.74	1.29	1.64	0.71	N/A		
3	1.28	2.19	1.74	2.37	1.54	1.22	N/A	
2	1.50	2.64	2.19	3.21	2.37	2.66	1.22	N/A
1	1.71	3.1	2.64	4.04	3.21	4.1	2.66	1.91

with such placement. It might also require more shield nets than available. A more advantageous approach is to constrain them into a regularly shaped region, e.g., a rectangle, as shown in Fig. 4(b). This can be implemented by enumerating all gates connected to target nets, and then creating a floorplan group to constrain their relative placements. The location of this floorplan group is chosen to remain as close to its original placement to reduce the impact on performance. The optimal dimensions of this floorplan group are found by extracting all gates and nets involved into a sub-layout where only these gates and nets are placed and routed.

The comparator is used to detect the attack by comparing the shield signal from upper layer and another copy from lower layer. So the comparator nets should also be protected like target nets because if these nets are tampered to maintain a static value the testability of the shield nets will be compromised. Hence, the comparator gates (green) are constrained in a floorplan group besides the target block as shown in Fig. 4(d).

Unlike target nets, we divide the gates connected to shield nets into two separate floorplan groups: *shield nets driver group* and *shield nets load group* as shown in Fig. 4(c). Our proposed shield net identification metric ensures that the performance overhead due to our constrained floor-planning is minimal. Both shield nets driver group and load group (blue) are constrained at opposite ends of the expected shielding area (target and comparator block) as shown in Fig. 4(d), so that routing of shield nets crosses the target area and therefore provides vertical protection from milling/probing. The

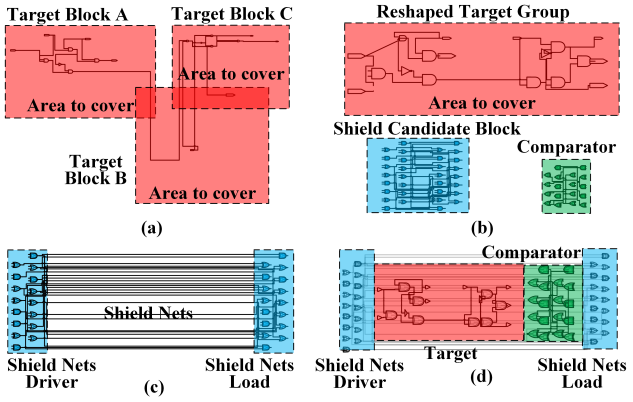

Fig. 4: (a) Irregular blocks (red) of sensitive target nets. (b) Reshape the sensitive target blocks to one regular rectangle block (red), shield candidate block (blue), and comparator block (green). (c) Shield gates (blue) are divided into shield nets driver block and shield nets load block. (d) Shield gates are placed surrounding the target and comparator blocks which will be covered by shield nets.

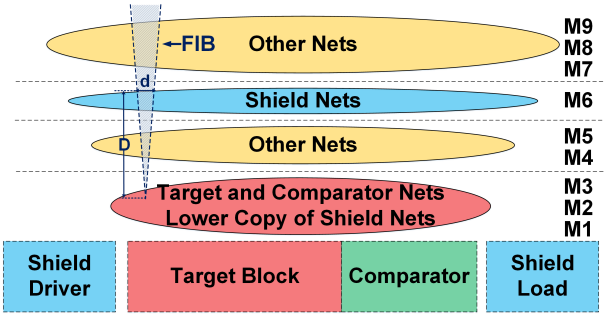

Fig. 5: Routing layer constraints for target and shield nets.

shield nets load group should be placed at the comparator's side. So that the received signals from shield nets could be compared in the comparator.

3.5 Routing Constraints

In addition to creating floor-planning constraints, wire-routing constraints are also very important to protect the device against probing attacks with large aspect ratio FIB. Aspect ratio of FIB is defined as the ratio between depth D and diameter d, as shown in Fig. 5, of a milled hole and is an important measure of FIB performance [13]. A larger aspect ratio results in a milling hole of smaller diameter on the top-most exposed layers, and therefore has less impact on the protective circuitry. Section 3.3 has revealed that routing shield wires on metal 6 and target wires under metal 4 can maximize the protection of shield against probing attack since it requires more advanced FIB with large aspect ratio to implement the attack without cutting off any net. Further, routing target nets in lower layer can also increase the coverage from other non-shield internal function nets in the design. In this paper, we route shield nets on M6 (M9 is the top layer), target nets and comparator nets under M4 to get an optimal protection as shown in Fig. 5. Further, another copy of shield nets are also routed under M4 to be compared with the genuine shield net from M6. To avoid design rule violations, part of shield nets have to be routed under M6, and part of target and comparator nets have to be routed on M4.

3.6 Exposed Area (EA) Calculation

To assess the design's vulnerability to probing attacks, [13] proposed a metric by calculating the exposed area (EA) of the design to probing attacks. The complementary part is the milling exclusion area (MEA). Fig. 6 shows how the exposed area (EA) can be found for any given target wire and covering wires on higher layers which are capable of projecting the milling exclusion area. Assuming the white region is the targeted wire at lower layer of a layout and the green and purple regions are the covering wires at upper layers above the targeted wire, the shaded region is the milling exclusion area (MEA), which indicates that if the milling center falls in this area then one of the covering wire (purple or green) will be completely cut off by the cone shape milling hole. Hence, the complement area of MEA is the desired exposed area that will not cause any cut-off of covering wires. The exposed area can vary according to the different aspect ratio of FIB, since the diameter of the holes milled by FIB with different aspect ratio is different. Larger exposed area in the

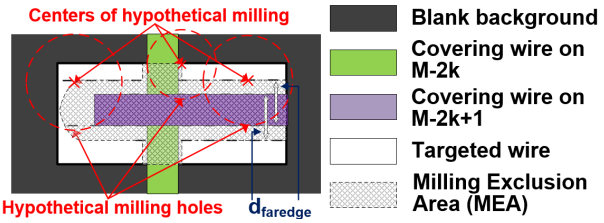

Fig. 6: Top-down view of a layout.

design represents more vulnerable to probing attacks.

4. EVALUATION

In this section, the proposed FIB-aware anti-probing physical design flow is evaluated to find out how efficient the design flow can be and how much area in the design is vulnerable to probing attacks. For this purpose, layout of Advanced Encryption Standard (AES) and Data Encryption Standard (DES) encryption modules are chosen for the evaluation of the proposed design flow.

4.1 Implementation of Proposed Design Flow

The DES and AES modules used are from OpenCores [14]. They are described in register-transfer level (RTL) code and synthesized to gate-level netlists using Synopsys *Design Compiler* with Synopsys SAED 32nm technology library. The layout of AES and DES modules are generated and constrained using Synopsys *IC Compiler*. The only asset in the AES and DES modules is the encryption key (128 bits for AES and 56 bits for DES) which is hardcoded in the design.

Gates connected to target nets and key memory cells are grouped and reshaped into a rectangular target block as shown in Fig. 7 (red). In addition, a 64-bit comparator is inserted in the AES design and a 32-bit comparator is inserted in the DES design. Comparator gates are also grouped and reshaped into a rectangular block besides target gates block as shown in Fig. 7 (green). 64 nets in AES module and 30 nets in DES module, which meet all requirements of shield metrics, are identified as shield nets for both designs. Therefore, in AES module, 64 driver gates and 64 load gates connected to the shield nets are reshaped into two groups respectively and placed at the opposite ends of target and comparator block as shown in Fig. 7 (blue). Target nets, comparator nets, and shield nets copy are constrained in the reshaped target and comparator block and routed under M4 as discussed in Section 3.5. Most shield nets are routed on M6 to provide

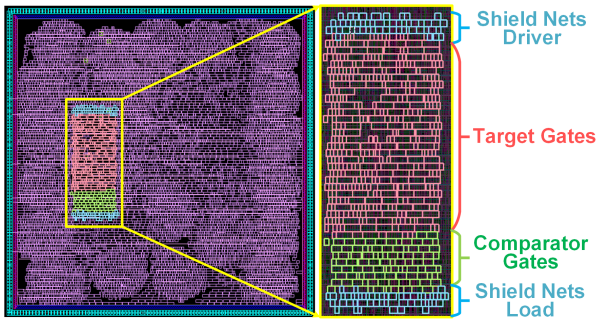

Fig. 7: Grouped and reshaped target gates, comparator gates, and shield gates in AES.

Table 3: Overhead of shielded design in AES and DES.

	Timing	Power	Area	Routing
AES	5.95%	5.92%	2.29%	16.89%
DES	0.00%	15.20%	4.66%	39.55%

coverage. Table 3 shows the timing, power, area and routing overhead of the shielded design compared to original design without any constraints. The area overhead is only 5% for both AES and DES designs since all shield nets are already contained in the design and the main area overhead is from the inserted comparator.

4.2 Exposed Area (EA)

To compare the shielding performance of our proposed internal shield approach and conventional active shield approach, an imaginary active shield is placed on the top-most layer (M9) of both original AES and DES designs for exposed area evaluation as illustrated in Section 3.6. Original nets on the top-most layer are removed. The space between active shield wires is the stipulated M9 pitch size in the SAED32nm technology library. Fig. 8(a)(c) show the percentage of reduced exposed area for active shied designs and internal shield designs compared to original designs in AES (Fig. 8(a)) and DES (Fig. 8(c)). The reduced exposed area is calculated as the exposed area difference between the new design and the original design over the exposed area of original design. As the FIB aspect ratio increases, the exposed area for all designs also increases since the milling exclusion area will decrease as d_{faredge} decreases with more advanced FIB. By using our proposed anti-probing design flow, the exposed area can be reduced by 90% when FIB aspect ratio is 1. Even with the most advanced FIB, the exposed area is still reduced by 78% for AES and 55% for DES. However, for the conventional active shield designs, the exposed area for both AES and DES are reduced by less than 18%, which is very ineffective. Fig. 8(b)(d) shows the percentage of exposed target wires, which is defined as target wires that have non-zero exposed area. Some wires can be fully protected, i.e. they don't have any exposed area. From Fig. 8(b)(d), with the most advanced FIB, 80% of wires from internal shield AES design and 70% of wires from internal shield DES design are fully protected (i.e., no exposed area). By contrast, the original design and active shield design has less than 40% and 50% of wires being fully protected respectively.

5. CONCLUSION

In this paper, we propose the FIB-aware anti-probing physical design flow, which incorporates two security critical steps in the conventional physical design flow. The floor-planning and routing of the design are constrained to provide coverage on asset nets. Evaluations on AES and DES modules show that the total vulnerable exposed area to probing attack of the anti-probing design decreases by 90% compared to original design. In addition, the area overhead is less than 5% for both designs, which can be totally ignored in an SoC. In future work, we will apply our anti-probing design flow to SoCs which contain more types of asset needing to be protected against probing attacks.

6. REFERENCES

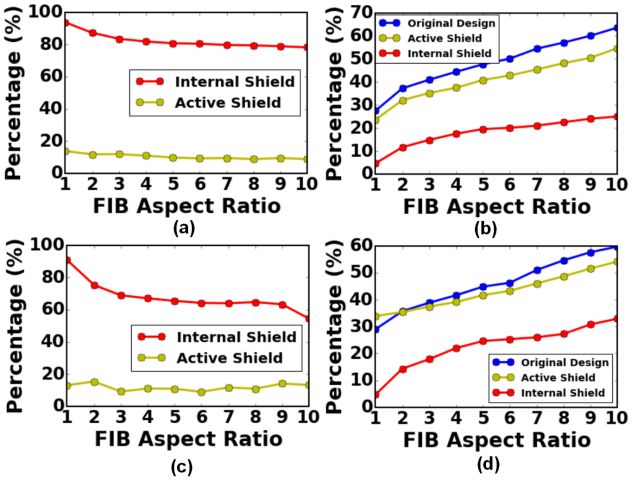

Fig. 8: (a) The percentage of reduced exposed area in AES. (b) The percentage of exposed target wires in AES. (c) The percentage of reduced exposed area in DES. (d) The percentage of exposed target wires in DES.

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

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

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

[4] 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," *Journal of Materials Science: Materials in Electronics* 25 (2): 587-595, 2014.

[5] J. Cioranesco, J. Danger, T. Graba, S. Guilley, Y. Mathieu, D. Naccache, X. 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.

[6] 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 *Computational Intelligence and Security (CIS)*, 2012 Eighth International Conference on, pp.530-533, 17-18 November 2012.

[7] S. Manich, M. Wamser, and G. Sigl, "Detection of probing attempts in secure ICs," in Proc. *2012 IEEE Int. Symp. Hardware-Oriented Secur. Trust*, Jun. 2012, pp. 134-139.

[8] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Advances in Cryptology-CRYPTO* 2003, D. Boneh, Ed.Berlin, Germany: Springer, 2003, pp. 463-481.

[9] H. Wang, Q. Shi, D. Forte, M. Tehranipoor, "Probing Attacks on Integrated Circuits: Challenges and Research Opportunities", in IEEE Design & Test, Volume: 34, Issue: 5, Oct. 2017.

[10] ARM Inc., "Building a secure system using TrustZone Technology." accessed Jul. 13, 2017. [Online].

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

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

[13] Q. Shi, N. Asadizanjani, D. Forte, and M. Tehranipoor, "A layout-driven framework to assess vulnerability of ICs to microprobing attacks," in Proc. IEEE Int. Symp. *Hardware Oriented Secur. Trust*, May 2016, pp. 155-160.

[14] http://opencores.org

[15] L. Goldstein, E. Thigpen, "SCOAP: Sandia Controllability/Observability Analysis Program", *Proceedings of 17th Design Automation Conference*, Minneapolis, MN, June 1980, pp. 190-196

[16] http://www.techinsights.com/

[17] R. Cole, J. Yakura, "Integrated circuit protection device and method," WO Patent No. WO/1997/036326. February, 10, 1997