A Chip Architecture for Compressive Sensing Based Detection of IC Trojans

A CHIP ARCHITECTURE FOR COMPRESSIVE SENSING BASED DETECTION OF IC TROJANS

Yi-Min Tsai[†]*, Keng-Yen Huang*[†]*, H. T. Kung*[*]*, Dario Vlah*[*]*, Youngjune L. Gwon*[*]*, and Liang-Gee Chen*[†]

[†]*Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan*
[*]*School of Engineering and Applied Sciences, Harvard University, USA*
[†]{ymtsai, kyhuang, lgchen}@video.ee.ntu.edu.tw [*]{htk, dario, gyj}@eecs.harvard.edu

ABSTRACT

We present a chip architecture for a compressive sensing based method that can be used in conjunction with the JTAG standard to detect IC Trojans. The proposed architecture compresses chip output resulting from a large number of test vectors applied to a circuit under test (CUT). We describe our designs in sensing leakage power, computing random linear combinations under compressive sensing, and piggybacking these new functionalities on JTAG. Our architecture achieves approximately a $10\times$ speedup and $1000\times$ reduction in output bandwidth while incurring a small area overhead.

***Index Terms*—** Compressive sensing, IC Trojan, CS-JTAG, measurement generator

1. INTRODUCTION

Fabless integrated circuit (IC) manufacturing has become a mainstream trend today, as many companies choose to outsource the manufacturing of their IC designs to wafer foundries. While this outsourcing model may provide advantages in manufacturing cost and an access to advanced fabrication facilities, there is an increasing concern about the security of these externally manufactured chips. It is generally more difficult to enforce security measures with external manufacturers which are under separate management, and as a result there is less assurance on these chips being free from possible malicious attacks during the fabrication process.

A Trojan [1–3] is malicious circuitry implanted in, for example, a CPU or encryption IC. Trojans may be a small addition to a normal circuit and may remain dormant until triggered by a special signal. During the incubation period it could be especially difficult to detect Trojans, given that there will be no functional difference between Trojan-free and Trojan-embedded circuits.

In the literature, there are three basic premises studied for a Trojan detector: a) timing of the circuit path could be slower because of supplementary Trojan gates [4]; b) Trojan will inevitably draw some static power [5]; and c) physical structure of the circuit is altered [6]. When the Trojan is not on the critical path, however, it is difficult to notice the timing modification. And finding out the circuit alteration can be impractical due to costly, destructive inspection. Therefore, detecting the power consumption difference induced by a Trojan can represent an attractive alternative [7–11]. This approach belongs to a class of detection methods called side channel analysis. It is the approach this paper takes.

Ideally, a Trojan can be detected by observing the leakage power difference between a circuit-under-test (CUT) and the corresponding Trojan-free circuit. The total leakage power of a circuit with the gate-level details can be modeled and described statistically [11, 12]. However, because of fabrication process variation [13,14], the leakage current distribution varies from gate to gate even with the same input states. This means that the Trojan power consumption could be hidden in process variation.

To combat the issue of process variation, statistical methods, such as [11], involving a large number of test vectors will be needed. In *DISTROY* [7], we have proposed an I/O-efficient method to discover revealing test vectors that can distinguish a Trojan-embedded circuit from a Trojan-free circuit. The approach relies on the assumption that such test vectors are rare, *i.e.*, they are *sparse*. Thus, by using compressive sensing (CS) [15, 16], which exploits signal sparsity, we can efficiently find the revealing test vectors from a large candidate pool (*i.e.*, the test vector space) without incurring excessive chip I/O.

In this paper we describe a chip architecture for a compressive sensing based IC Trojan detection method under the *DISTROY* framework [7]. We explore architecture issues for various CS regularity conditions such as the restricted isometry property (RIP) [17]. Our goal is to provide an IC design house or a foundry with a realistic and low-cost Trojan detection infrastructure.

Our architecture leverages the commonly used Joint Test Action Group (JTAG) boundary scan standard [18]. We therefore name this architecture compressive sensing-JTAG, or *CS-JTAG*. CS-JTAG provides not only the original function of JTAG but also compressive encoding capabilities for efficient detection of possible malicious implants.

The rest of the paper is organized as follows. We explain

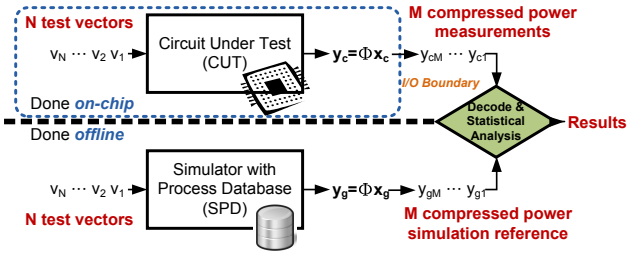

Fig. 1. Chip architecture for CS-based IC-Trojan detection.

the CS-based Trojan detection in Sec. 2. Sec. 3 presents the proposed chip architecture. We discuss our design in Sec. 4 and evaluate the performance of our implementation in Sec. 5. Sec. 6 concludes the paper.

2. COMPRESSIVE SENSING BASED TROJAN DETECTION

2.1. Compressive Sensing (CS) Fundamentals

Compressive sensing [15, 16, 19] is a new signal processing paradigm that can acquire a sparse signal representation with linear random projections. It allows encoding a signal with sparsity constraints and recovering the signal with only few compressed measurements. More precisely, supposed that we can represent a signal $\boldsymbol{x}$, an $N \times 1$ vector, as a sparse approximation using a basis $\boldsymbol{\Psi}$:

$$\boldsymbol{x} \approx \sum_{i=1}^{K} s_i \psi_i = \boldsymbol{\Psi s} \tag{1}$$

where the ψ_i are basis functions, with $K \ll N$. Then $\boldsymbol{x}$ is said K-*sparse* and compressible. CS encoding computes an $M \times 1$ measurement $\boldsymbol{y}$ of compressive measurements, each random linear combination of K-sparse $\boldsymbol{x}$ using an $M \times N$ measurement matrix $\boldsymbol{\Phi}$:

$$\boldsymbol{y} = \boldsymbol{\Phi x} \tag{2}$$

Matrix $\boldsymbol{\Phi}$ is also called a *sensing matrix*. It has randomly chosen entries. To reconstruct the signal $\boldsymbol{x}$ from $\boldsymbol{y}$, we use an underdetermined linear system given by Eq. 3 where there are more unknowns (N) than equations (M). The CS theory states that the signal can be reconstructed, with M being as small as $cK\log(N/K)$ for a small constant c, using ℓ_1-norm minimization:

$$\min \|s\|_{\ell_1} \quad s.t. \quad \boldsymbol{y} = \boldsymbol{\Phi\Psi}^{-1}\boldsymbol{s} \tag{3}$$

Note that a number of optimization algorithms can be used such as compressive sampling matching pursuit (CoSaMP) and iterative soft-thresholding (IST). The CS theory shows that more measurements result in more accurate recovery.

2.2. CS-based Trojan Detection Approach

Our proposed CS-based approach (Fig. 1) reduces I/O bandwidth requirement while maintaining the same testing quality. First, we avoid inputting test vectors by generating them on-chip. Second, chip output is reduced from N power measurements to M random linear combinations of these measurements, where $M \ll N$. The off-chip Trojan detector then recovers the most significant power variations from the M power measurements.

An on-chip test vector generator (TVG) generates N test vectors, $v_1, v_2, ..., v_N$ that are applied to the CUT. The corresponding N leakage power measurements, $x_{c1}, x_{c2}, ..., x_{cN}$ ($\boldsymbol{x_C}$) are then compressed into M linear combinations, $y_{c1}, ..., y_{cM}$ ($\boldsymbol{y_C}$), on-the-fly through multiplication by a sensing matrix $\boldsymbol{\Phi}$. The simulation reference ("gold" measurement), $\boldsymbol{x_G}$ is also multiplied by the same $\boldsymbol{\Phi}$ to get $\boldsymbol{y_G}$.

We perform CS reconstruction of $\boldsymbol{x_C}-\boldsymbol{x_G}$ off-chip based on $\boldsymbol{y_C}-\boldsymbol{y_G}$. Since the vector $\boldsymbol{x_C}-\boldsymbol{x_G}$ is expected to be sparse, the required number of measurements M can be small. Based on the recovered $\boldsymbol{x_C}-\boldsymbol{x_G}$, statistical analysis proposed in [7] is then applied to examine suspicious chips and make the final decision about whether or not the CUT is deemed to be Trojan-embedded. Note that CS reconstruction can be performed off-chip with highly parallel computing platform such as GPU. In this paper, we focus on the on-chip sensing architecture.

3. CS-JTAG ARCHITECTURE

Our chip architecture aims at providing an automatic self-examination scheme for discovering malicious Trojans without complicated interfaces. As shown in Fig. 2, the on-chip detecting architecture includes a TVG, a leakage power sensor (LPS), a measurement generator (MG), and the CS-JTAG controller governing the original JTAG controller. Except LPS which is analog, all the other circuits are digital. The following subsections describe the data flow and major modules in detail.

3.1. Data Flow and Schedule

Fig. 3 depicts the data flow and work scheduling of the architecture. The TVG applies each test pattern, v_i, to the CUT per clock cycle. After the CUT settles down, the LPS measures each leakage power sample, x_{ci}, corresponding to its test pattern. The MG performs matrix-vector multiplication to form compressed measurements y_{ci} while receiving x_{ci}. Note that all y_{ci} are produced at the same time after the last leakage power data x_{cN} is received. Each y_{ci} is outputted afterwards.

3.2. CS-JTAG Controller

The JTAG standard was originally developed for boundary scan and internal device tests during chip production. Fig. 4

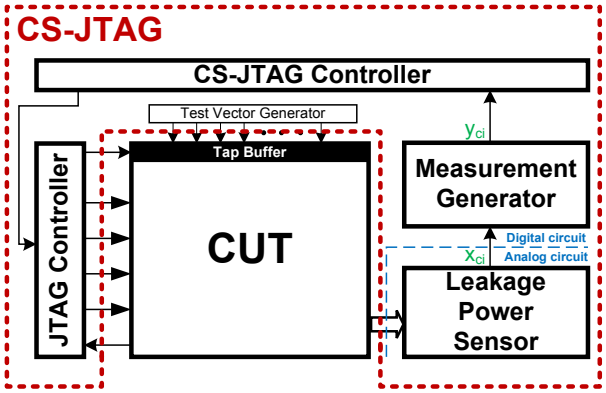

Fig. 2. On-chip Trojan detection architecture.

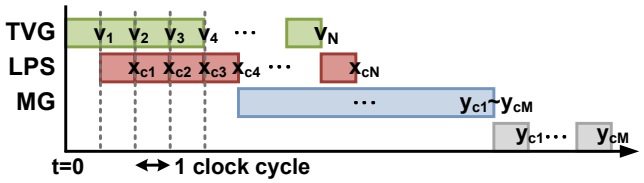

Fig. 3. Data flow and work scheduling.

depicts the original JTAG controller and how it interfaces to our CS-JTAG controller. There are three input ports to control JTAG controller: *clock* decides the debug operating frequency, *enable* controls the state transition of output control signals, and *input* allows the user to insert the test vector and the expected result one bit per cycle. The JTAG operating stage can be divided into three steps, *get*, *shift*, and *set*. Each step individually manages the register scan chain around the CUT to get the output data, shift the test vector along the scan chain, and update the input register buffers.

Similarly, our proposed architecture is also a three-step process, parallel test vector insertion, leakage power sensing, and measurement generation. It implies that the JTAG ports can be used as a chip infrastructure for our Trojan-detection purposes. To this end, CS-JTAG manipulates a new *Trojan enable* signal to control the state transition of the JTAG. The consequent JTAG output control signals then are utilized to control the three new modules. As a result, no additional port is needed for the CS-JTAG architecture and the CS-JTAG controller is simplified to merely enable the JTAG controller to support CS related operations.

3.3. Leakage Power Sensor

Since ICs typically operate at a fixed voltage, we can refer to leakage current I_o and leakage power P_L interchangeably. As illustrated in Fig. 5, we duplicate the circuit current by a current mirror to make $I' = I_o$. By fixing the resistance R,

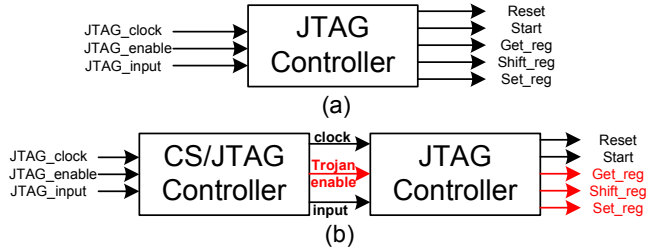

Fig. 4. (a) Original JTAG interface and control signal. (b) Inserting CS-JTAG controller at the input side of JTAG controller, and using the Trojan enable signal to trigger the new behavior of JTAG.

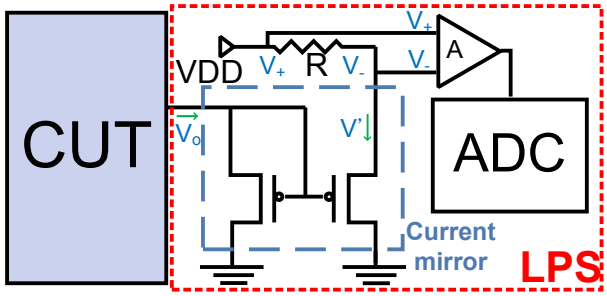

Fig. 5. The leakage power sensor (LPS) with ADC.

the voltage drop $V_+ - V_-$ can be represented as in Eq. 4.

$$V_+ - V_- = I' \cdot R = I_o \cdot R \tag{4}$$

$$P_L = VDD \cdot I_o = VDD \cdot \frac{V_+ - V_-}{R} \tag{5}$$

An analog-to-digital converter (ADC) is applied in LPS to calculate the voltage drop, as suggested by [20]. After knowing $V_+ - V_-$, the circuit leakage power can be determined as depicted in Eq. 5. For calculating P_L of each test vector, the ADC is needed to be fast enough to provide power results in every cycle. However, the leakage current is usually too small to drive ADC. An opamp A is designed to increase the current driving capability and accelerate power settling time.

3.4. Measurement Generator

The measurement generator comprises circular shift registers (CSRs), linear feedback shift registers (LFSRs), and selective adders (Fig. 6). The compressed measurement y_{ci} is computed in a shift and accumulate manner, using a Bernoulli sensing matrix with coefficients $\phi_{ij} \in \{+1, -1\}$. The original leakage power data is sparse in time domain and the Bernoulli sensing matrix is incoherent with our representative basis.

Note that each y_{ci} is the inner product of a row vector in $\mathbf{\Phi}$ and $\boldsymbol{x_C}$. That is, $y_{c1} = \phi_{11}x_{c1} + \phi_{12}x_{c2} + ... + \phi_{1N}x_{cN}$, $y_{c2} = \phi_{21}x_{c1} + \phi_{22}x_{c2} + ... + \phi_{2N}x_{cN}$ and so on. The partial

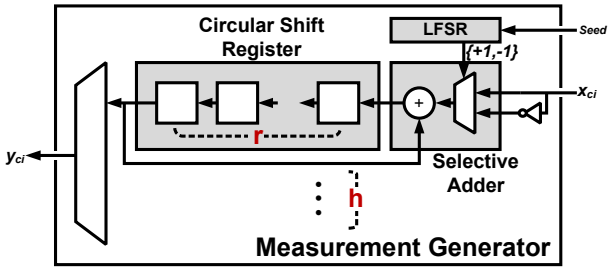

Fig. 6. Measurement generator (MG).

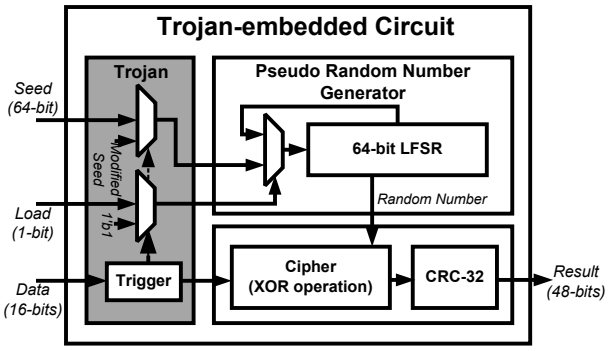

Fig. 7. The Trojan-embedded circuit under test.

sum of y_{ci} is updated when receiving a new x_{ci}. LFSRs are used to generate selection bits for summand (i.e., x_{ci} or $-x_{ci}$) multiplexers. The current selected summands are added to the corresponding accumulated partial sums circularly shifted to the selective adders. After computing all M measurements, MG sequentially outputs the results stored in the CSRs.

4. DESIGN CONSIDERATIONS

4.1. Operating Frequency Aspects

We denote the sampling frequency of LPS by f_{LPS}, and the operating frequency of MG, CUT by f_{MG}, f_{CUT} respectively. To measure stable power data, the f_{LPS} should not be smaller than f_{CUT}. For instance, if f_{CUT} is 100 MHz, f_{LPS} should be at least 100M samples/s. We denote the maximum f_{CUT} by $f_{CUT_{max}}$ and consider the following cases.

• **High capability LPS, $f_{LPS} \geq f_{CUT_{max}}$.** In this case, the CS-JTAG achieves maximum throughput and shortest latency for generating y_{ci}, provided that $f_{MG} \geq f_{LPS}$. Given that the testing cannot be performed faster than $f_{CUT_{max}}$, there is no need to design f_{LPS} to be larger.

• **Low capability LPS, $f_{LPS} < f_{CUT_{max}}$.** The CS-JTAG bottleneck is determined by f_{LPS}.

Note that the design constraint $f_{MG} \geq f_{CUT}$ holds in both cases. However, the area-efficiency may increase by selecting a higher f_{MG} (discussed in 4.2). Depending on the target testing frequency and throughput requirement, we then design the architecture with suitable f_{LPS} and f_{MG}.

4.2. Area Aspects

The most area-intensive part lies in MG (Fig. 6). Thus it is significant to reduce the area of MG. Here defines a frequency ratio $r = \lfloor f_{MG}/f_{LPS} \rfloor$ and a parallelism ratio $h = \lceil M/r \rceil$. After receiving a power data x_{ci} from LPS, MG has r clock cycles to calculate all M partial sums until the next x_{ci} comes. In each cycle, MG calculates h partial sums in parallel (i.e., MG calculate partial sums of $y_{c1} \sim y_{ch}$ at the first clock cycle and those of $y_{ch+1} \sim y_{c2h}$ at the second, etc.). Thus, there are total h CSRs, h LFSRs, and h selective adders in MG. Each CSR consists of r registers. We now discuss three factors influencing circuit area.

• **Number of Measurements, M.** MG requires M registers for buffering partial sums of compressed measurements. Hence, the area cost is linearly proportional to M.

• **Frequency Ratio, r.** With M fixed, low frequency ratio results in high parallelism of MG. This indicates more selective adders and LFSRs are required.

• **Bitwidth of Measurements, $BW(y_{ci})$.** The bitwidth of y_{ci}, defined as $BW(y_{ci})$, directly affects the gate counts of registers in MG. We can reduce $BW(y_{ci})$ but still maintain compression quality. x_{ci} is firstly biased by a reference value in the middle range of leakage power. Because the Bernoulli random matrix has coefficients in $\{+1, -1\}$, we then reduce $BW(y_{ci})$ depending on the coefficient distribution.

5. PERFORMANCE ANALYSIS

5.1. Circuit Under Test with Embedded Trojan

As a test circuit for CS-JTAG, we designed an encryption circuit (Fig. 7) including an deliberately-inserted Trojan for performance evaluation purposes. The circuit uses a LFSR to generate pseudo random number for an XOR-based cipher and produces the corresponding cyclic redundancy check code. The Trojan is activated while triggered by a specific input pattern. It then changes the seed and load control of the pseudo random number generator, resulting in an unreliable cipher. The Trojan-free circuit can be obtained directly by removing the Trojan part and connecting the corresponding ports. In the simulation, we set the 64-bits seed to be a constant and use exhaustive (N=2^{16}) test vectors for 16-bits data input. The maximum testing frequency $f_{CUT_{max}}$ is assumed to be 200MHz. All circuits, including CS-JTAG and circuit under test, are synthesized and simulated in 90nm general purpose process.

5.2. Synthesis Results

Fig. 8 shows the synthesis results of area-speed trade-off under various design factors. We can reduce area cost by either choosing a smaller M or a lower f_{LPS}.

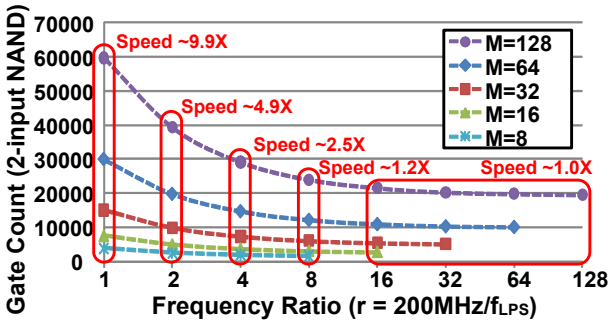

Fig. 8. Gate count versus frequency ratio under different M.

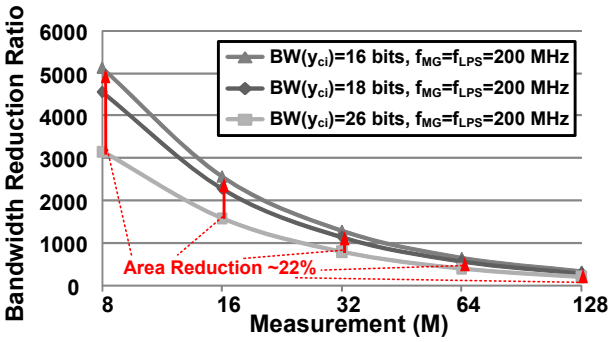

Fig. 9. Bandwidth reduction ratio.

Without CS, the output time dominates the system performance. It requires about 3.27 ms to finish testing all 2^{16} vectors if f_{LPS}=f_{MG}=200MHz. With CS, the testing time is reduced to 0.33 ms, implying about 10× speedup. Note that as f_{LPS} is extremely slow, power sensing time becomes critical and there is no gain by adopting CS.

Fig. 9 shows the output bandwidth reduction ratio. When y_{ci} is 26-bits and M is 32, the proposed approach achieves about 780× improvement against the baseline approach. The improvement is even higher if y_{ci} is 16-bits and M is 8. Besides, it results in about 22% area reduction of MG while the bitwidth of y_{ci} is reduced from 26 bits to 16 bits.

5.3. Simulation Results

In the CS-based Trojan detection scheme, we only need to find the most revealing vectors. With its the largest-first decoding property, CS decodes the largest abnormal power consumption values first. In our design experiments, we set sparsity K to be less than 8, M to 64, N to 2^{16}, and y_{ci} to 16 bits. f_{LPS} is set to 200MHz. Note that the system should follow M=$cK\log(N/K)$ constraint to have correct reconstructions. The total area cost of CS-JTAG is about 32K at 200MHz. The Trojan-embedded circuit is about 2K gates with Trojan being approximately 0.18K gates (about 8% of the total area of the CUT). We then perform gate-level circuit power analysis using simulation CAD tools given this design specification. We show the leakage power distribution for the Trojan-embedded circuit and the Trojan-free circuit, respectively, under process variation. Note that here we only illustrate leakage power values for 1024 consecutive test vectors (Fig. 10). The proposed architecture is further adopted to encode the sensed power data of CUT on-line into compressed measurements. By applying our previous work, *DISTROY* [7], the compressed measurements are off-line decoded to discover several vectors that can separate the leakage power distribution of the Trojan-free circuit from that of the Trojan-embedded circuit. Therefore, the statistical analysis in *DISTROY* can then determine the false positive rate and the detection rate for detection decisions. In the future, we plan to verify the proposed architecture and the testing procedure through the hardware implementation.

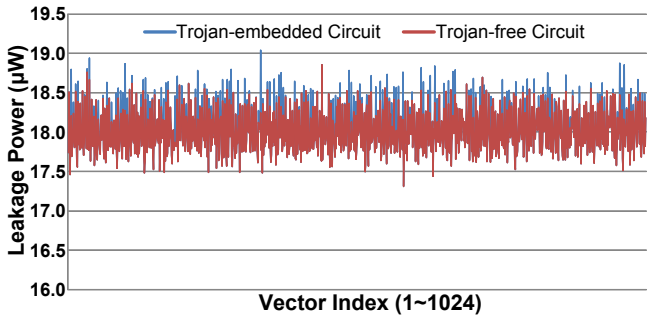

Fig. 10. Simulated leakage power values.

6. CONCLUSION

Compressive sensing based detection of IC Trojans is capable of identifying test vectors which reveal Trojans, without subjecting to excessive amounts of chip I/O. In this paper, we have shown a chip architecture to realize the required chip functionalities such as sensing of leakage power and computing of random projections. In addition, we have shown an approach leveraging the existing JTAG architecture. Based on these results, we conclude that chip realization of compressive sensing based IC Trojans detection is feasible.

Acknowledgment

This material is in part based on research sponsored by Air Force Research Laboratory under agreement number FA8750-10-2-0180. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

7. REFERENCES

[1] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *Design Test of Computers, IEEE*, vol. 27, no. 1, pp. 10–25, Jan. 2010.

[2] J. Rajendran, E. Gavas, J. Jimenez, V. Padman, and R. Karri, "Towards a comprehensive and systematic classification of hardware Trojans," in *Circuits and Systems (ISCAS), Proceedings of IEEE International Symposium on*, June 2010, pp. 1871–1874.

[3] R.S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware Trojan: Threats and emerging solutions," in *High Level Design Validation and Test Workshop, IEEE International*, Nov. 2009, pp. 166–171.

[4] Yier Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *Hardware-Oriented Security and Trust, IEEE International Workshop on*, June 2008, pp. 51–57.

[5] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," in *Security and Privacy, IEEE Symposium on*, May 2007, pp. 296–310.

[6] Y. Alkabani and F. Koushanfar, "Consistency-based characterization for IC Trojan detection," in *Computer-Aided Design (ICCAD), IEEE/ACM International Conference on*, Nov. 2009, pp. 123–127.

[7] Y.-L. Gwon, H.-T. Kung, and D. Vlah, "DISTROY: Detecting integrated circuit Trojans with compressive measurements," in *USENIX Workshop on Hot Topics in Security (HotSec)*, 2011.

[8] S. Narasimhan, Dongdong Du, R.S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia, "Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach," in *Hardware-Oriented Security and Trust (HOST), IEEE International Symposium on*, June 2010, pp. 13–18.

[9] M. Banga and M.S. Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans," in *VLSI Design, International Conference on*, Jan. 2009, pp. 327–332.

[10] D. Shamsi, P. Boufounos, and F. Koushanfar, "Noninvasive leakage power tomography of integrated circuits by compressive sensing," in *Low Power Electronics and Design, IEEE/ACM International Symposium on*, Aug. 2008, pp. 341–346.

[11] Y.-L. Gwon, H.-T. Kung, D. Vlah, K.-Y. Huang, and Y.-M. Tsai, "Statistical Screening for IC Trojan Detection," in *Circuits and Systems (ISCAS), Proceedings of IEEE International Symposium on*, 2012.

[12] R. Fernandes and R. Vemuri, "Accurate estimation of vector dependent leakage power in the presence of process variations," in *Computer Design, IEEE International Conference on*, Oct. 2009, pp. 451–458.

[13] Hongliang Chang and Sachin S. Sapatnekar, "Full-chip analysis of leakage power under process variations, including spatial correlations," in *ACM Proceedings of Design Automation Conference*, 2005, pp. 523–528.

[14] Rajeev Rao, Ashish Srivastava, David Blaauw, and Dennis Sylvester, "Statistical estimation of leakage current considering inter- and intra-die process variation," in *ACM Proceedings of International Symposium on Low Power Electronics and Design*, 2003, pp. 84–89.

[15] E.J. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[16] D.L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[17] E.J. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[18] "IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture," *IEEE Std 1149.7-2009*, pp. c1–985, Oct. 2010.

[19] R.G. Baraniuk, "Compressive Sensing [Lecture Notes]," *Signal Processing Magazine, IEEE*, vol. 24, no. 4, pp. 11 –121, July 2007.

[20] Y.-D. Jeon, Y.-K. Cho, J.-W. Nam, K.-D. Kim, W.-Y. Lee, K.-T. Hong, and J.-K. Kwon, "A 9.15mW 0.22mm2 10b 204MS/s pipelined SAR ADC in 65nm CMOS," in *Custom Integrated Circuits Conference (CICC), IEEE*, Sept. 2010, pp. 1–4.