**QUEEN'S
UNIVERSITY
BELFAST**

Security Analysis of Hardware Trojans on Approximate Circuits

Dou, Y., Yu, S., Gu, C., O'Neill, M., Wang, C., & Liu, W. (2020). Security Analysis of Hardware Trojans on Approximate Circuits. In *GLSVLSI '20: Proceedings of the 2020 on Great Lakes Symposium on VLSI* (pp. 315-320). ACM.

Published in:

GLSVLSI '20: Proceedings of the 2020 on Great Lakes Symposium on VLSI

Document Version:

Peer reviewed version

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

Copyright 2020 ACM. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Security Analysis of Hardware Trojans on Approximate Circuits

Yubin Dou
Nanjing University of Aeronautics
and Astronautics, Nanjing
Nanjing, China
douyubin@nuaa.edu.cn

Shichao Yu
Queen's University Belfast
Belfast, United Kingdom
syu08@qub.ac.uk

Chongyan Gu
Queen's University Belfast
Belfast, United Kingdom
C.Gu@qub.ac.uk

Maire O'Neill
Queen's University Belfast
Belfast, United Kingdom
m.oneill@ecit.qub.ac.uk

Chenghua Wang
Nanjing University of Aeronautics
and Astronautics, Nanjing
Nanjing, China

Weiqliang Liu
Nanjing University of Aeronautics
and Astronautics, Nanjing
Nanjing, China
liuweiqliang@nuaa.edu.cn

ABSTRACT

Approximate computing, for error-tolerant applications, provides trade-offs for computations to achieve improved speed and power performance. Approximate circuits, in particular approximate arithmetic circuits, directly affect the performance of a computing system. Hence, approximate circuit designs have been extensively studied. However, security issues of approximate circuits have been ignored. Moreover, hardware Trojans have been found in fabricated chips in manufacturing industry chains by untrusted foundries. Hardware Trojans could affect the functionality of approximate circuits under very rare circumstances with inconsiderable footprints. In this paper, hardware Trojan insertion methods based on signal transition probability are utilized to investigate and evaluate the security threats in approximate circuits. A approximate low-part-or-adder (LOA) adder is utilized as an example and analyzed in the paper. The evaluation results show that with the increase of the number of approximation modules, the approximate LOA adder is more possible to be inserted hardware Trojans than the exact LOA adder.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems; Redundancy; Robotics**; • **Networks** → **Network reliability**.

KEYWORDS

approximate circuits, hardware Trojan, transition probability

ACM Reference Format:

Yubin Dou, Shichao Yu, Chongyan Gu, Maire O'Neill, Chenghua Wang, and Weiqliang Liu. 2020. Security Analysis of Hardware Trojans on Approximate Circuits. In *Proceedings of the Great Lakes Symposium on VLSI 2020 (GLSVLSI '20)*, September 7–9, 2020, Virtual Event, China. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3386263.3407591>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '20, September 7–9, 2020, Virtual Event, China

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7944-1/20/09...\$15.00

<https://doi.org/10.1145/3386263.3407591>

1 INTRODUCTION

The Moore's law is approaching its limit, which indicates that conventional computing techniques are difficult to provide efficient computing performance for computing systems according to the restriction of power consumption. Approximate computing is emerging as a promising paradigm for energy efficient designs [10]. It is suitable for high speed and low power nanoscale integrated circuits (ICs) for many applications where high accuracies are not required and intrinsic errors are acceptable [4].

Approximate circuits, based on basic arithmetic elements at logic level, attract many research attention. Thanks to approximation, approximate circuits, computing systems can be faster, more compact and consume less power. However, the development of approximate circuits nowadays mostly focuses on its design and application level, ignoring its security problem. Approximate circuits can bring different security issues to hardware [1] [17]. For example, it is difficult to distinguish the final output error whether they contain malicious falsification due to the uncertainty and unpredictability of intrinsic errors during approximate execution. Nowadays, third parties have been widely involved in IC designs and manufacturing in supply chains. From the design of IC chips to the process of manufacturing, these outsourcing companies can insert purposely additional malicious circuit logic units to circuits, also known as "hardware Trojan". Hardware Trojans have been recognized as one of the most harmful attacks in hardware circuits [8]. They can be added to the original circuit in the form of bypass circuits, and can be used to destroy circuits or steal data under the control of an attacker. Adversaries can maliciously change or add additional functionality to an IC using hardware Trojans [14].

Since approximate circuits are available to an untrusted foundry in fabrication, hardware Trojans could be inserted, and then the inexact design of the approximate circuits may bring more opportunities for attackers. Therefore, it is necessary to develop an approach to analyze the vulnerability of approximate circuits by hardware Trojan insertion. Transition probability, which represents the probability for an input port's transition from 1 to 0 or 0 to 1, is a key factor for hardware Trojan insertion. Generally, to reduce a Trojan's activation probability, its triggers are connected to the nets with low transition probabilities. Hence, we will evaluate the security vulnerability in approximate circuits using transition probability approaches. The transition probabilities of normal (exact)

circuits are also utilized for a comparison. In this paper, this is the first time, as authors' best knowledge, to present a fair analysis of security vulnerabilities in approximate circuits using the transition probability method for hardware Trojans. More specifically, the main contributions of this paper are as follows:

- The vulnerabilities of approximate circuits are investigated and evaluated through analyzing the transition probability of hardware Trojan insertion.
- A comparison of transition probabilities between two different types of circuits, approximate circuits and normal circuits, is presented.
- A flow for comparative analysis of approximate circuit and exact circuit is proposed.

The rest of the paper is organized as follows. Section 2 explains the background of this work. Section 3 and Section 4 discuss the details of Hardware Trojan, approximate circuit and transition probability. In the Section 5, an flow for analyzing the vulnerability of approximate circuits is presented. An experimental setup and results are analyzed. Section 6 concludes the paper.

2 BACKGROUND

The globalization of integrated circuits provide opportunities for attackers to hack/damage hardwares. Every outsourcing company related to IC manufacturing can become a potential adversary to insert malicious modifications, referred as hardware Trojans. Currently, the investigation of vulnerabilities of ICs has raised global attention. It is also worth to evaluate the potential threats if approximate circuits are inserted hardware Trojan. Hence, the security of approximate circuits is an important aspect although it has been neglected.

2.1 Approximate computing

Approximate computing can improve computing performance by losing the accuracy of calculation results [7]. "Inexact Design-Beyond Fault-Tolerance" [5] and "Good Enough Computing" [15] were all calling for the importance of approximate calculations. Therefore, it has attracted many research efforts, especially for approximate arithmetic circuits, which include adders, multipliers and dividers. Arithmetic circuits play an important role in the processor [12], in which arithmetic circuits directly affect the performance and power consumption of the whole computing system. Hence, it is expected to achieve higher speed and lower power consumption as well as satisfied the use of fault-tolerant applications. All these motivated the fast development of approximate arithmetic circuit designs [2]. Design methods for approximate circuits mainly include logic reduction and voltage based probability CMOS techniques [3]. The probability CMOS techniques can save power by supplying a normal voltage to the most significant bit and a low voltage to the least significant bit. It reduces power consumption and has only a small impact on final results. The technique allows the adder to save power. However, the technology is difficult to implement. The cost is high, and the error is difficult to control. Therefore, most of the approximate computing circuits are based on the approach of logic reduction, which includes logic gate reduction or replacement with different gate elements. Although the approximate circuit is computationally more efficient and consumes less

energy, the current development is mostly reflected in the design level, while ignoring its security issue.

Moreover, hardware Trojans and other security threats have shown up to attack the approximate circuits. [1] pointed that a hardware Trojan can be inserted during the approximate synthesis flow, and it is difficult to be detected due to approximate fault tolerance. [17] introduced a potential attack, malicious modification, on approximate adder by deliberately manipulating the adder's inputs to continuously generate erroneous outputs to activate error correction code (ECC) or fault tolerant process more than usual. It has been shown that the correlation between the output and power for the adder with 50% errors is higher than that of 25% errors. [13] discussed that approximate devices might include extra circuits to control the level of approximation. However, the extra circuits are a natural target for hardware Trojans insertion. It also stated that the error-detection and self-adaptation circuits needed to support the approximated computing paradigm can offer a new channel to leak information. Therefore, the security of approximate circuits confront challenges and hardware Trojans have negligible effect on approximate circuits. Hence, it is necessary to analyze the vulnerabilities of approximate circuits in hardware Trojan insertion, and in order to better analyze the approximate circuits, both the approximate and exact circuits are compared in the paper. Analyzing the vulnerabilities of approximate circuits provides prevention for the insertion of hardware Trojans during chip manufacturing and design stages. The analysis can be used as a guidance for hardware Trojan detection on approximate circuits. Furthermore, from the comparative analysis of both the exact and approximate circuits, an approximate structure with less influence on the circuit can be derived.

2.2 Hardware Trojan

Hardware Trojan is a malicious modification of the circuit by an attacker. It normally includes trigger circuit and payload circuit. The trigger part is the activation condition of a hardware Trojan. It is usually connected to a circuit node with a lower transition probability. The payload part is responsible for executing a malicious attack on the target circuit when the trigger part is activated. Hardware Trojans can control, modify, disable or monitor circuit components and communications. Fig. 1 shows an example of the Trojan-infected circuit. The trigger circuit activates the subsequent payload circuit when trigger circuit receives the specified signal, including the specified input or internal status. The payload circuit can leak information or destroy circuit after activation. [1] indicated that the payload circuit could be connected to the output of an approximation circuit so that the error of the result exceeds satisfied error tolerance for cognitive applications. Hardware Trojans are closely related to transition probabilities. Trojans are usually added to nodes with low transition probabilities and at the same time some hardware Trojan detection methods are also based on transition probabilities. [18] proposed an approach to utilize transition probability for automated hardware Trojans insertion to a circuit. In this paper, the same transition probability calculation will be employed.

Figure 1: An example of the Trojan-infected

Figure 2: Transition probability diagram

2.3 Transition Probability

A circuit is a combination of a number of logic gates. The gate is the most basic digital unit that has the ability to allow or block the flow of digital information. Typically, a gate circuit contains one or more inputs that produce an output that is a function of the input value. The state of each input port can be 0 or 1, and the probability of 1 and 0 is equal. At the same time, the state of the output can also be 0 and 1, but the probability is not equal. The probability calculation requires the truth tables of gate-level circuits. For example, for an XOR gate, the input ports are A and B, and the output is C. Only if A, B signals are opposite, C terminal output is 1. For signals a and b, the probability of $(a=0 \text{ and } b=1)$ is $(p_{a0}, p_{b1}) = (1/2, 1/2)$, which leads to the probabilities of output signal $(c=0)$ and $(c=1)$ are $p_{c0} = 1/2$ and $p_{c1} = 1 - p_{c0} = 1/2$, respectively. Hence, the switching probability from “0” to “1” or “1” to “0” can be defined as (1):

$$p_{tran} = p_{c1} * p_{c0} \quad (1)$$

Fig. 2 shows an example of transition probability diagram for a hardware Trojan based computational logic circuit. Assume that the probabilities of input 0 or 1 are both 1/2. By calculating the node probability of each gate, the state probability of the nodes in the circuit can be obtained successively, and then the probability of the output is obtained. Finally, the flip probability of the final gate is derived as $69/128 * (59/128) = 0.2484$. When using this topology of the circuit to obtain the flip probability of a node, the geometric distribution can be used to model one flip. The geometric distribution is a discrete distribution, and the probability function can be expressed as follows:

$$P(n) = p * (1 - p)^n \quad (n = 0, 1, \dots) \quad (2)$$

The function in (2) indicates that the node will flip at the $(n + 1)$ th clock cycle. (p_0, p_1) are utilized to indicate the probabilities

that a circuit node is 0 and 1. The probability calculation process follows three principles: 1) nodes and nodes are independent of each other, 2) satisfying coincidence $p_0 + p_1 = 1$, and 3) the input port probability is $(1/2, 1/2)$. In the paper, (2) is utilized to find nodes with low probability for approximate circuits to estimate and compare their security with normal circuits.

3 APPROXIMATE CIRCUITS

Most of the current research on approximate circuits has focused on approximate arithmetic units. Further, among the approximate arithmetic unit, the study of approximate adders has attracted a lot of attention. The reasons are: 1) Adders are the basic components of many arithmetic units, such as addition, subtraction, multiplication, and division, which need adders to perform fundamental data operations. 2) Adders are also the basic building blocks of digital systems, and they affect the performance of digital systems. Therefore, approximate adders have been widely researched. In this paper, we will use an approximate adder and an exact adder as the target circuit for our analysis.

Faster and more energy-efficient adders can be achieved by designing shorter carry chains or using specific bits. A number of different approximate adders has been developed. An overview and classification of current approximate adders is presented in [9]. A survey [6] has also reviewed different types of approximate adders.

A low-part-or-adder (LOA) design is chosen as an example to illustrate the approximate adder. The idea of LOA is to execute exact calculation for the more important bits in an output and inexact calculation for the less important bits in the output. As shown in Fig. 3, a k -bit LOA consists of two parts: an m -bit exact adder and an n -bit inexact adder. The revised LOA adder shown in Fig. 3 is adopted as

Figure 3: The revised LOA adder structure [9].

our target analysis circuit. The exact part is made up of n exact 1-bit adders connected in parallel, and the approximate part is formed by m approximate 1-bit adders connected in parallel. For a k -bit adder, $k = n + m$, where k is the k -bit adder, n is the n -bit precise part and m is the m -bit approximate part. In the experiment, n traditional 1-bit full adders are used to form the precise part, and m approximate 1-bit adders proposed in [16] are used to form the approximate part. The traditional 1-bit full adder and the approximate 1-bit adder are shown in Fig. 5. a, b and c are the inputs. S and C are the output. Unlike the exact adder of Fig. 5 (a), the approximate adder of Fig. 5 (b) is an NAND-based design, which has three NAND gates and one inverter. In normalized CMOS logic gate, NAND has the lowest delay [11]. Hence, the NAND-based approximate adder

Figure 4: A flow for the analysis of an approximate circuit.

has an improvement of 29% for the critical path delay [16], which means the operation speed of the approximate adder in Fig. 5 (b) could be much faster than that in Fig. 5 (a).

4 TRANSITION PROBABILITY ANALYSIS

In this paper, an operational flow is proposed to compare and analyze the vulnerabilities of approximate circuits. The proposed flow uses the transition probability as an analysis approach. The approximate adder, the revised LOA design, is used as our target circuit. A vulnerability analysis is also performed on the LOA adders in different structures.

Generally, in an approximate circuit, some modules are approximated, and some modules remain as exact. In particular, when the approximate circuit is designed, it needs to be based on the fault tolerance of the application, such as the LOA design in Fig. 3, where the values of k , m , n are determined by the actual application. Therefore, an approximate circuit structure needs to be considered in the vulnerability analysis.

In logic circuits, the importance of each path is different. For example, in the arithmetic circuit, the most significant bit is an important path, since the most significant bit has the highest weight. Inserting a Trojan to the important path is more destructive than that of the less important path. Therefore, in order to analyze the vulnerability of the circuit after approximation, transition probability can be utilized to compare and analyze the important path of the approximate circuit and the exact circuit.

To address this, a flow as shown in Fig. 4 is proposed to compare and analyze the vulnerabilities of approximate circuits. First, hardware description languages such as VHDL or Verilog are utilized to implement both approximate and exact circuits. EDA tools, such as synopsys, are used to generate the netlists of both approximate and exact circuits. The physical design tool calculates the transition probability of each node in the synthesized netlists for both the

Figure 5: Adders. (a) 1-bit exact adder and (b) 1-bit approximate adder.

approximate and exact circuits. Finally, the transition probability is utilized to perform a vulnerability analysis from two perspectives: approximate circuit structure and important path.

5 EXPERIMENTAL RESULTS

5.1 Critical Path Analysis

The critical path analysis uses physical synthesis tools to calculate the transition probability of each node of both the approximate and the exact circuits. The transition probabilities of critical path nodes are compared and analyzed. For the arithmetic unit, the significant output bit belongs to a critical path. An inexact output will be generated when hardware Trojans are inserted to the more significant path.

Figure 6: The transition probabilities of an 8-bit approximate adder.

Fig. 6 shows that the transition probability distribution of the critical path of an approximate 8-bit adder. The change of the transition probability of the most significant bit, the second significant bit and the Cout of the approximate module increases from 0 to 6. Similarly, Fig. 7 presents the transition probability distribution of the critical path of an approximate 16-bit adder, including the four output paths with the highest output weight and Cout, and the approximate module is increased from 4 to 12. It can be seen that the transition probability of the critical output path of the approximate adder decreases with the increase of the number of approximate modules. In Fig. 6, we can see that when there are only

Figure 7: The transition probabilities of a 16-bit approximate adder.

two approximation modules, the transition probability does not change much. However, when the approximate module increases to six, the transition probability decreases significantly. The 16-bit approximate adder in Fig. 7 has similar results. When the number of approximation modules is four, the transition probability does not change much. When it is increased to twelve, the transition probability drops significantly, and the transition probability of the most significant bit becomes almost 0. Such nodes are particularly suitable for Trojan insertion. The transition probability is related to the circuit structure and logic gate, and the introduction of approximation modules (an approximation module is an approximate adder based on NAND gate). The more the approximate modules, the greater the structural change of the circuit, which brings the more change to the transition probability.

5.2 Approximate Circuit Structure Analysis

The approximate circuit structure analysis will investigate the vulnerabilities of the approximate circuits in different structures, and exact circuits will be involved for comparison. In the experiment, the approximate circuit is developed and the EDA tool is used to obtain the transition probability, and then the data analysis is performed in MATLAB.

Fig. 8 shows a scatter diagram of the transition probabilities of the approximate LOA circuit in different approximate parameters (n) and exact LOA adder circuit. The blue dot line represents the transition probability of the exact LOA adder circuit and the red triangle line illustrates the transition probability of the approximate LOA circuit, the revised LOA circuit, in different parameters. Fig. 8 (a), (b), (c) present the transition probabilities of an 8-bit approximate LOA adder in different approximate parameters, $n = 2, 4, 6$, respectively, compared to that of an 8-bit exact LOA adder. It can be seen that with the increase of the number of approximation modules (n), the number of low transition probabilities of the approximate LOA adder (i.e., the number of red dots) is more than that of the exact LOA adder (i.e., the number of blue triangles). Hence, it means that the approximate LOA adder circuit is more vulnerable to hardware Trojan insertion with the increase of the

number of approximate modules. Fig. 8 (d), (e), (f) show the transition probabilities of a 16-bit approximate LOA adder in different approximate parameters, $n = 4, 8, 12$, respectively, compared to that of a 16-bit exact adder. For the 16-bit approximate LOA adder, the more the number of approximate modules, the higher the possibility of hardware Trojan insertion. Especially in Fig. 8 (f) the number of low transition probabilities of the approximate LOA adder with 12 approximate modules is much more than that of the exact LOA adder without any approximation. The value of some transition probabilities of the approximate LOA adder are even close to 0, which has the highest risk to hardware Trojan insertion. Compared to the exact LOA adder, the transition probability of the last node of the approximate LOA adder is lower, which has a potential vulnerability of hardware Trojan insertion. Through the analysis of transition probability of circuit design, potential vulnerabilities of the LOA circuits can be revealed. Hence, it provides an effective approach for designers to detect potential threats in the design of approximate circuits.

6 CONCLUSION

In this paper, we have investigated and evaluated the security vulnerabilities of approximate adder circuits, particularly the approximate LOA adder circuit. A flow is presented for the comparative analysis of approximate circuits with exact circuits. Hardware Trojan insertion methods based on signal transition probability analysis have been utilized to investigate and evaluate the vulnerabilities of approximate circuits. The approximate LOA adder is used as an example for security analysis. The experimental results show that the number of low transition probabilities of the approximate LOA adder is more than that of the exact LOA adder. With the increase of the number of approximate modules, the number of low transition probabilities of the approximate LOA adder is more than that of the exact LOA adder. For the approximate LOA adder, the more the number of approximation modules, the more possible the approximate LOA adder inserted hardware Trojans. In a future work, the security vulnerabilities of different approximate designs, for example, approximate multiplier and approximate divider, will be investigated, not only in hardware Trojan insertion but also other attacking techniques.

ACKNOWLEDGMENTS

This work was partly supported by the EPSRC (EP/N508664/-CSIT2), the National Natural Science Foundation China (61871216) and the Nature Science Foundation of Jiangsu Province (BK20151477).

REFERENCES

- [1] Sheikh Ariful Islam. 2019. On the (In)security of Approximate Computing Synthesis. *arXiv e-prints*, Article arXiv:1912.01209 (Dec 2019), arXiv:1912.01209 pages. arXiv:1912.01209 [cs.CR]
- [2] V. Camus, M. Cacciotti, J. Schlachter, and C. Enz. 2018. Design of Approximate Circuits by Fabrication of False Timing Paths: The Carry Cut-Back Adder. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8, 4 (Dec 2018), 746–757. <https://doi.org/10.1109/JETCAS.2018.2851749>
- [3] Suresh Cheemalavagu, Pinar Korkmaz, Krishna Palem, and Lakshmi Chakrapani. 2020. A Probabilistic CMOS Switch and its Realization by Exploiting Noise. (02 2020).
- [4] J. Han and M. Orshansky. 2013. Approximate computing: An emerging paradigm for energy-efficient design. In *2013 18th IEEE European Test Symposium (ETS)*. 1–6. <https://doi.org/10.1109/ETS.2013.6569370>

Figure 8: The transition probability distribution of an approximate LOA adder with the parameters of (a) 8-bit, $k = 8$, $n = 2$, (b) 8-bit, $k = 8$, $n = 4$, (c) 8-bit, $k = 8$, $n = 6$, (d) 16-bit, $k = 16$, $n = 4$, (e) 16-bit, $k = 16$, $n = 8$, (f) 16-bit, $k = 16$, $n = 12$, respectively, compared to an exact LOA adder.

- [5] Cheng-Chian Lin Ing-Chao Lin, Yi-Ming Yang. 2015. High-performance low-power carry speculative addition with variable latency. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst* 23 (2015), 1591–1603.
- [6] Honglan Jiang, Cong Liu, Leibo Liu, Fabrizio Lombardi, and Jie Han. 2017. A Review, Classification, and Comparative Evaluation of Approximate Arithmetic Circuits. *Emerg. Technol. Comput. Syst.* 13, 4, Article 60 (Aug. 2017), 34 pages. <https://doi.org/10.1145/3094124>
- [7] Andrew B. Kahng and Seokhyeong Kang. 2012. Accuracy-configurable adder for approximate arithmetic designs. In *Dac Design Automation Conference*.
- [8] He Li, Qiang Liu, and Jiliang Zhang. 2016. A survey of hardware Trojan threat and defense. *Integration* 55 (2016), 426 – 437. <https://doi.org/10.1016/j.vlsi.2016.01.004>
- [9] Weiqiang Liu, Chongyan Gu, Gang Qu, and Máire O'Neill. 2018. *Approximate Computing and Its Application to Hardware Security*. Springer International Publishing, Cham, 43–67. https://doi.org/10.1007/978-3-319-98935-8_3
- [10] Weiqiang Liu, Fabrizio Lombardi, and Michael Shulte. 2020. A Retrospective and Prospective View of Approximate Computing. *Proc. IEEE* 108, 3 (2020), 394–399.
- [11] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi. 2017. Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing. *IEEE Trans. Comput.* 66, 8 (2017), 1435–1441.
- [12] V. Mrazek, Z. Vasicek, L. Sekanina, H. Jiang, and J. Han. 2018. Scalable Construction of Approximate Multipliers With Formally Guaranteed Worst Case Error. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26, 11 (2018), 2572–2576.
- [13] Francesco Regazzoni, Cesare Alippi, and Ilia Polian. 2018. Security: The Dark Side of Approximate Computing?. In *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD'18)*. IEEE, 1–6.
- [14] Walid Saad, Anibal Sanjab, Yunpeng Wang, Charles A. Kamhoua, and Kevin A. Kwiat. 2017. Hardware Trojan Detection Game: A Prospect-Theoretic Approach. *CoRR* abs/1703.07499 (2017). arXiv:1703.07499 <http://arxiv.org/abs/1703.07499>
- [15] A. K. Verma, P. Brisk, and P. Ienne. 2008. Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design. In *2008 Design, Automation and Test in Europe*. 1250–1255. <https://doi.org/10.1109/DATE.2008.4484850>
- [16] Haroon Waris, Chenghua Wang, and Weiqiang Liu. 2019. High-performance approximate half and full adder cells using NAND logic gate. *IEICE Electronics Express* 16 (02 2019). <https://doi.org/10.1587/elex.16.20190043>
- [17] Pruthvy Yellu, Novak Boskov, Michel A Kinsky, and Qiaoyan Yu. 2019. Security Threats in Approximate Computing Systems. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI*. ACM, 387–392.
- [18] S. Yu, W. Liu, and M. O'Neill. 2019. An Improved Automatic Hardware Trojan Generation Platform. In *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. 302–307. <https://doi.org/10.1109/ISVLSI.2019.00062>