Character Reassignment for Hardware Trojan Detection

Noah Waller, Hunter Nauman, Derek Taylor, Rafael Del Carmen, Jia Di
Computer Science and Computer Engineering Department
University of Arkansas
Fayetteville, USA
{ndwaller, hjnauman, dt003, rddelcar, jdi}@uark.edu

***Abstract*— With the current business model and increasing complexity of hardware designs, third-party Intellectual Properties (IPs) are prevalently incorporated into first-party designs. The use of third-party IPs increases security concerns related to hardware Trojans inserted by attackers. Previous work on Golden Reference Matching focuses on matching with all entries within a single Golden Reference Library (GRL) containing whitelisted and blacklisted functionalities. This paper presents two new Golden Reference Libraries, Champion GRL and Functionality GRL, which were introduced along with coarse-grained and fine-grained asset reassignment to soft IPs and GRL entries in order to improve matching accuracy while simultaneously saving computational resources.**

Keywords—hardware Trojan, asset, structural checking, golden reference matching

I. INTRODUCTION

Due to economic considerations, it is no longer financially feasible to design every component of an Integrated Circuit (IC) in-house. Therefore, 1st party vendors contract 3rd parties to design certain components. By doing so, the integrity of the overall soft IP can be compromised through the insertion of hardware Trojans into a 3rd party component, posing threats to these applications. A hardware Trojan can be defined as "a malicious, intentional modification of a circuit design that results in undesired behavior when the circuit is deployed" [1]. Sample payloads of hardware Trojans include the leaking of cryptographic keys and denial-of-service attacks for devices.

One area of research for detecting hardware Trojans is at the soft IP level. Soft IPs are Register-Transistor-Level (RTL) code or other gate-level netlist. One strategy for detecting hardware Trojans using gate-level netlists was developed in [2] where the use of natural language processing and statistical analysis distinguished the "naturalness" of a circuit against the "unnaturalness" of a hardware Trojan. Another strategy using gate-level netlists was introduced in [3]. Machine learning was used on net testability and netlist structural features to detect the instance of a possible Trojan.

Different from the research for soft IPs mentioned previously, the Golden Reference Matching method in [4] uses RTL code, rather than gate-level netlists. Golden Reference Matching breaks apart RTL code into components and primary ports. Signals are labelled using assets which describe the functionality of the overall soft IP. Once assets have been assigned to the unknown soft IP, it is compared against a Golden Reference Library (GRL). This GRL contains a collection of entries that are known to be either Trojan-free or Trojan-infested. Once compared all entries within the GRL, the soft IP with the highest match to an entry is categorized. If the unknown soft IP best matches against a Trojan-infested entry, then the IP likely contains a Trojan, and vice versa.

To decrease computational resource usage while preserving categorization of soft IPs, a subset of entries within the GRL were taken and used as champion entries to be used in a newly developed Champion GRL. This champion entry is considered the best entry of a functionality and is used initially in matching, where the unknown soft IP is given the functionality of the highest match within the Champion GRL. Due to the limited number of designs within the Champion GRL, external assets were generalized into 10 categories and are reassigned to the soft IP to increase matching between functionalities. Once a functionality is assigned to the soft IP, it is then matched to designs only within its functionality, leading to the Functionality GRL, a GRL organized based upon functionality.

The rest of this paper is organized as follows. Section II will cover background information on assets, Structural Checking (SC), and Golden Reference Matching with a Golden Reference Library. Section III will cover the design and implementation of asset reassignment through coarse-grained and fine-grained applications. Section IV will provide examples of soft IPs to prove the effectiveness of the improved matching process and section V will conclude the paper and provide details on future work.

II. BACKGROUND

A. Assets

1) Overview: Assets provide labels to both primary ports as well as internal signals of a soft IP about their purpose in the context of the design hierarchy. Each signal can have multiple asset labels assigned to improve the description of the overall design. There are two categories of assets, internal assets and external assets.

2) Internal Assets: Internal assets are for internal signals but can be used to describe primary port signals as well. Most internal assets used in the Structural Checking (SC) tool were developed in [5] and [6]. Most internal assets are automatically assigned, while a small subset assigned mannualy.

3) External Assets: External assets are for primary ports in a soft IP. They are all manually assigned to using the SC tool.

This work is supported by National Science Foundation Award 1703602

The majority of external assets were developed in [4] and [5] and are grouped into 5 categories: *Data*, *Timing*, *System Control*, *Specific System Control*, and *Miscellaneous*. Each category encompasses signals contributing to the domain of the given category. Assets falling in the *Data* category pertain to the flow of data through a circuit, whereas assets located inside of the *Timing* category pertain to the timing of a circuit. *System Control* and *Specific System Control* assets relate to the control of a circuit. *Miscellaneous* assets refer to any other types of signals that may be defined within a circuit but do not fall into the other categories.

4) Asset Filtering and Patterns: Asset Filtering, introduced in [7], is used to propagate assigned assets of a signal through all signals connected to it. Propagating assets allows the tool to find correlations between signals as well as conflicting asset assignments. Once asset filtering is completed, an asset trace is created for every signal in a design. The collection of asset traces is stored inside of an asset pattern. The asset pattern is broken down into 6 characteristics: external assets assigned to input ports, internal assets assigned to input ports, external assets assigned to output ports, internal assets assigned to output ports, external assets assigned to internal signals, and internal assets assigned to internal signals.

B. Golden Reference Matching

1) Overview: The Golden Reference Matching process compares the asset pattern of an unknown soft IP against the asset pattern of an entry within the GRL and determines whether it contains a Trojan. For each entry in the GRL, the algorithm behind the matching process calculates a percent match against the unknown soft IP and the soft IP's functionality is based on the highest match.

2) Asset Reassignment: Reassignment of a specific asset label to a more generalized label is utilized as introduced in [8]. This idea stems from work completed in [4], where a specific asset can be matched with a generic counterpart as two signals could theoretically be the same, but due to certain assets not having been introduced in earlier stages, a more generic asset was assigned to the given signal.

3) Statistical Matching: Statistical matching was added to the SC Tool's matching algorithm in [8]. Assets included in a single characteristic of numerous GRL entries should have a lower matching weight compared to assets that are found within a subset of entries. The matching weight of an asset is is the probability of an asset not being found in a given entry. Once all asset weights are calculated, an average asset weight is then calculated based on the sum of the matched asset weights divided by the total number of matched assets within a given characteristic. Equation (1) demonstrates the calculation for the total weight of a characteristic. After the average asset weight is determined for the characteristic, it is divided by the sum of all characteristics' average asset weights. This quotient is then converted into a percentage based on the sum of the 6 characteristic's average asset weight within the GRL.

4) Golden Reference Library: The GRL is a collection of soft IPs retrieved from Trust-Hub [9] and OpenCores [10]. They are labeled as Trojan-free (Whitelist) or Trojan-infested (Blacklist). An unknown IP that matches with a blacklisted functionality is flagged as potentially containing a Trojan.

$$Weight_{char} = \frac{Characteristic_{char}.AverageAssetWeight}{\sum_{i=A}^{F} Characteristic_i.AverageAssetWeight} * 100 \quad (1)$$

III. Methodology and Implementation

A. Overview

The SC matching process described in Section II leads to an inefficient use of computational resources. As the tool has improved with the GRL containing over 160 entries for comparison, the matching process itself has not been altered. As a result, when a new soft IP is introduced to the tool, it is compared against all entries, leading to an increase in computational and memory resources. To address this, a Champion GRL consisting of a single entry from every whitelisted functionality was incepted. When an unknown soft IP is matched using this Champion GRL, a whitelisted functionality is assigned to the soft IP based on the highest percentage match. Once the functionality is determined, the original GRL is portioned into distinct functionalities, so the soft IP can match against entries of the same functionality.

B. Champion Golden Reference Library Matching

1) Champion Golden Reference Library: To establish a general functionality for the unknown soft IP, GRL entries are inspected manually, and a subset of entries are included in this separate library. An entry needs to contain both multiple asset traces and specific assets within each trace, but there must be a compromise to ensure an unknown design can be matched with confidence.

2) Coarse-Grained Asset Reassignment: Because of entry limitations of the Champion GRL, a coarse-grained-to-coarse-grained asset reassignment set was introduced. Due to the fine-grained comparisons of the unknown IP and the Champion GRL entries, top GRL matches have a lower percentage with the soft IP when compared against a single design within the same functionality. Coarse-grained matching resembles asset reassignment and is utilized on external characteristics.

a) Asset Set One: Asset Set One contains a list of 10 generalized external asset categories encompassing all external assignments, as listed in Table 1. Two new external assets, *SPECIFIC_CONTROL* and *EXTRA*, were created for coarse-grained asset reassignment, as not all fine-grained assets enjoy a generic equivalent. This version of asset reassignment is used on both the soft IP and the Champion GRL entries to produce the highest possible percentage match between assets and functionality. During testing, Asset Set One was not able to correctly identify soft IPs with similar functionalities. Because of this and the possibility of a soft IP's highest percent match going below a given threshold, a second asset set was created.

TABLE I. ASSET SET ONE

Asset Category	Assets
DATA_COMPUTATIONAL	DATA_COMPUTATIONAL, DATA_MEMORY, DATA_SENSITIVE
DATA_COMMUNICATION	DATA_COMMUNICATION, DATA_PERIPHERAL
DATA_ENCRYPTION	DATA_ENCRYPTION, KEY
SYSTEM_TIMING	SYSTEM_TIMING, SUBSYSTEM_TIMING
STATUS	STATUS, READY, DONE, BUSY, HOLD, COUNT, WAIT, COMMUNICATION_STATUS
SYSTEM_CONTROL	SYSTEM_CONTROL, ENABLE, SET, RESET, EXECUTE, READ, WRITE, INTERRUPT, SELECT, HANDSHAKING, SHIFT, LOAD, MODE, INSTRUCTION
ADDRESS_SENSITIVE	ADDRESS_SENSITIVE, REGISTER
SPECIFIC_CONTROL	SPECIFIC_CONTROL, INTERRUPT_CONTROL, PERIPHERAL_CONTROL, REGISTER_FILE_CONTROL, COMMUNICATION_CONTROL, TIMER_CONTROL, CLOCK_CONTROL, COMMUNICATION_PROTOCOL, DATA_OP, MEMORY_OP, INTERRUPT_OP, PROGRAM_COUNTER_OP, BUS_CONTROL, LCD_CONTROL, LED_CONTROL, PHASE, DUTY_CYCLE
EXCEPTION_HANDLING	EXCEPTION_HANDLING, ERROR_HANDLING
EXTRA	EXTRA, CRITICAL, COMPONENT, STATE, UNKNOWN, UNUSED

b) Asset Set Two: Considering how the GRL is defined and how soft IPs are developed, certain assets are more common than any data asset in the GRL. Asset Set Two classifies each data asset into a new category. Assets within *STATUS* and *TIMING* are combined. Assets within the categories *SYSTEM_CONTROL*, and *SPECIFIC_CONTROL*, *EXCEPTION_HANDLING* are all combined.

To determine if Asset Set Two is required, the top two matches are needed. First, the top match is compared against a threshold of 40%. If this threshold is not met, the confidence is not high enough in regards with assigning functionality and the soft IP has its assets reassigned using Asset Set Two. If this threshold is met, the top match is compared with the second highest match. If the difference between the two matches is less than 15%, there is a possibility that the soft IP matches with different designs containing similar functionalities, which facilitates the use of Asset Set Two. If both criteria are met, matching within the Functionality GRL is performed.

C. Functionality Golden Reference Library Matching

1) Overview: As addressed in Section III.A, the original GRL matched an unknown soft IP against all entries within the library, regardless of functionality. The inclusion of the Champion GRL rendered the original GRL obsolete, so a new GRL was created. Separating the GRL into functionalities decreases resource demands during the process matching while simultaneously retaining matching percentages.

2) Fine-Grained Asset Reassignment: To facilitate GRL entry matching, fine-grained asset reassignment was conceived to increase the matching percentage of soft IPs with Functionality GRL entries. This scheme of asset reassignment contrasts with the others in that only the functionality GRL designs are reassigned, due to the unknown soft IP containing the most recent assets assigned to them. To establish which asset(s) need reassignment using the Functionality GRL's specific characteristic, the same characteristic of the unknown soft IP is used. Each asset within an asset trace is compared against one another. If both assets from the soft IP and GRL entry are the same, no asset reassignment is needed. If the two assets differ but are within the same asset category, the asset within the Functionality GRL's entry is reassigned to the soft IP's asset. If both assets are different and are not within the same asset category, the asset is not reassigned. A third asset set, named Asset Set Full, was added to accomplish this. This asset set contains 38 categories with each external asset grouped together based on similarities between one another.

IV. RESULTS AND ANALYSIS

To confirm the tool's ability to maintain correct functionalities with the changes made, results from [8] were used for evaluation. The tested IPs include BasicRSA-T200 and RS232-T700. In addition, a microcontroller was used to test the improved matching process.

A. Examples

1) BasicRSA and RS232 Modules: A Trojan-infested BasicRSA module, BasicRSA-T200 was used for comparison. This module contains a denial-of-service Trojan which disables encoding at the transmitter and decoding at the receiver. Asset Set One correctly identifies the IP as an encryption unit, however, Asset Set Full misidentifies it as an encryption unit. The second highest match, with a difference of 5.24%, identifies the module as containing a Trojan. The reasoning behind this change in functionality is the Functionality GRL contains few encryption unit-based entries. RS232, another Trojan-infested soft IP, was used for comparison between the two matching processes as well. This contains another denial-of-service Trojan on the transmitter, rendering the transmitter unable to receive communication after completion of a task. Both Asset Set One and Asset Set Two are needed for this IP as there are both Communication and Peripheral functionalities. With a difference of 12.38% between the two functionalities, it is not significant enough to determine the functionality. Asset Set Two determines that the functionality is Communication. Using Asset Set Full, the tool correctly classifies the module as containing a Trojan. This new method of matching decreases total memory usage by 35% and 14% between BasicRSA and the RS232 module, respectively.

2) Microcontroller: A larger microcontroller was tested to demonstrate improvements with the new matching process. This microcontroller, named Bus Interface, contains a ROM module, an SPRAM module, LED outputs, and a UART communication module.

Table II shows the final matching results between the statistical matching process and the new matching process.

Osch, PLL_Clock, Bus_Master, Vhi, Mux321, and Spr16x4c did not go through asset reassignment for Asset Set Two as their highest match were all above 40% and the difference between the top two matches were all higher than 15%. Osch is an oscillator and is correctly included in the Timing functionality, differing from its original functionality from statistical matching. Bus_Master is correctly assigned using both the statistical and new matching algorithms. PLL_Clock clock is an example of a subcomponent that was assigned the correct functionality using statistical matching but was not assigned correctly using the new matching method. This is due to current biases with the Champion GRL, where certain functionalities contain entries that are smaller than other entries, resulting in lower functionality matching results. Statistical matching incorrectly identifies Std_Counter as containing a Trojan while the new matching process correctly identifies the subcomponent as Trojan-free.

TABLE II. BUS INTERFACE MATCHING RESULTS

	Statistical Matching Process		**New Matching Process**	
Target IP	**Functionality**	**% Match**	**Functionality**	**% Match**
Bus_Interface_Top	Communication	35.02	Communication	20.56
Osch	Communication	34.88	Timing	28.79
PLL_CLK	Timing	79.18	Shift_Register	70.42
Vlo	Computational	99.16	Computational	98.98
Ehxpllj	Communication	52.91	Encryption_Unit	53.57
Bus_Master	Communication	69.46	Communication	87.66
SPRAM	Register_File	91.70	Communication	84.03
Inv	Computational	78.36	Computational	62.42
Rom16x1a	Register_File	66.77	Register_File	65.26
Vhi	Computational	95.52	Control_Generation	95.69
Fd1p3dx	Control_Generation	69.51	Decoder_Encoder	51.47
Mux321	Computational	61.34	Decoder_Encoder	96.60
Spr16x4c	Register_File	94.44	Register_File	94.27
RS232_Usr_Int	Communication	69.64	Communication	44.22
STD_FIFO	Register_File	66.15	Communication	79.11
Bus_Int	Register_File	74.85	Communication	74.67
Std_Counter	Trojan_Trigger	54.80	Computational	79.89
LED_Ctrl	Communication	60.80	Communication	49.97
PWM_16b	Register_file	67.06	Communication	62.87

As for the computational resource usage, the new method of matching decreases memory usage by 77%. This significant decrease in memory usage stems from the significant number of sub-level entries within the GRL that the new matching process categorizes, rather than comparing all entries.

V. CONCLUSION AND FUTURE WORK

Improvements in asset reassignment and the creation of the Champion GRL and the Functionality GRL enhance the efficiency of the matching process for the SC tool while also maintaining a high level of accuracy regarding functionality matching. The inclusion of a subset of GRL entries decreases memory usage by up to 77%. Unknown soft IPs that contain similar functionalities can be distinguished using multiple asset sets with relative accuracy. Microcontrollers are an example of soft IP that have a relatively low matching percentage due to the limited number of entries the GRL contains. Future work will continue to grow the list of functionalities as well as improve designs within the Champion GRL to decrease the use of Asset Set Two. The addition of new external assets can benefit the new functionality process as well by more effectively classifying unknown soft IPs.

REFERENCES

[1] M. Tehranipoor and C. Wang. 2012. Introduction to Hardware Security and Trust. Springer.J. Clerk.

[2] H. Shen, H. Tan, H. Li, F. Zhang and X. Li, "LMDet: A "Naturalness" Statistical Method for Hardware Trojan Detection," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 4, pp. 720-732, April 2018, doi: 10.1109/TVLSI.2017.2781423.

[3] C. H. Kok et al., "Net Classification Based on Testability and Netlist Structural Features for Hardware Trojan Detection," 2019 IEEE 28th Asian Test Symposium (ATS), Kolkata, India, 2019, pp. 105-1055, doi: 10.1109/ATS47505.2019.00020.

[4] Weaver, L., Le, T., & Di, J. (2016). Golden Reference Library Matching of Structural Checking for securing soft IPs. SoutheastCon 2016. doi:10.1109/secon.2016.7506737

[5] M. Hinds, J. Brady, and J. Di, "Signal Assets - a Useful Concept for Abstracting Circuit Functionality," presented at the Government Microcircuit Applications & Critical Technology Conference (GOMACTech), 2013.

[6] T. Le, J. Di, M. Tehranipoor, and L. Wang, "Tracking data flow at gate-level through structural," in 2016 International Great Lakes Symposium on VLSI (GLSVLSI), 2016, pp. 185-189.

[7] J. Yust, M. Hinds, and J. Di, "Structural Checking: Detecting Malicious Logic without a Golden Reference," Journal of Computational Intelligence and Electronic Systems, vol. 1, no. 2, p. 8, 2012.

[8] B. McGeehan, F. Smith, T. Le, H. Nauman and J. Di, "Hardware IP Classification through Weighted Characteristics," 2019 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 2019, pp. 1-6, doi: 10.1109/HPEC.2019.8916225

[9] H. Salmani, M. Tehranipoor, and R. Karri, "On Design vulnerability analysis and trust benchmark development", IEEE Int. Conference on Computer Design (ICCD), 2013.

[10] OpenCores. Available: http://opencores.org/