

Metrics for Analyzing Quantifiable Differentiation of Designs with Varying Integrity for Hardware Assurance

Adam G. Kimura¹, Steven B. Bibyk¹, Brian P. Dupaix¹, Matthew J. Casto², Gregory L. Creech¹

¹The Ohio State University - Department of Electrical and Computer Engineering (Columbus, OH)

²Wright Patterson Air Force Research Labs (Dayton, OH)

Contact Author Email: kimura.11@osu.edu

Abstract — *This work proposes an approach to quantifying the integrity of a questionable design by parsing the design into characteristic sub-domains: Logical Equivalence, Signal Activity Rate, Functional Correctness, Structural Architecture, and Power Consumption. Measurement techniques are reviewed for each domain which quantify deviation of the actual design away from the expected profiles. A novel method for quantifying the quality of reference used for expected profiles is also proposed. Expected profiles can incorporate a level of overdesign. Finally, the Design Integrity measuring techniques are applied to five Test Article (TA) cases that showed Error 2 TA to have the lowest integrity of 2.95/5 and the untampered TA containing the highest integrity of 5.00/5.*

Keywords — Trojan; integrity; trust; quantify; hardware; assurance; verification; metrics; reference, quality; profile

I. INTRODUCTION

A. The Rising Concern of Hardware Trust

As Integrated Circuit (IC) chips continue to advance in complexity, economics and time-to-market pressures are driving hardware developers into distributed design processes and complex supply chains. This has led to more opportunistic points in the design and manufacturing flow for error insertion by adversarial or dishonest agents inside a supplier. A hardware error is defined as any construct that causes deviation from the intended specification. Hardware errors are typically categorized as either faults or hardware Trojans. Hardware Trojans are inserted into the design with malicious intent to compromise a design's functionality and reliability. Other aims of hardware Trojans could be for granting control to an adversary for monitoring or stealing information. A fault is a quality control occurrence usually caused by poor fabrication processes; however faults are not typically malicious in nature.

With the globalization of the IC industry, hardware untrustworthiness (i.e. the concern for hardware error insertion) has become a growing issue as the Internet of Things continues to expand [1]. The rising concern for hardware trust has therefore led to the emergence of a new field of research to address these concerns - *Trusted Microelectronics*.

B. The Need for Trust Metrics

Developing a portfolio of metrics to quantify aspects of Trust such as design integrity and vulnerability are critical components to Trusted Microelectronics. Trust Metrics for quantifying design integrity could provide measurable insight into how closely the fabricated hardware from an untrusted foundry matches the original design by providing a distance measure for how far it deviated from it. Hardware vulnerability metrics could also be integrated into Computer-Automated Design (CAD) toolsets in order to grant quantifiable insight into various vulnerability mitigation strategies for improving design security. In addition, as standards for Trusted Microelectronics are developed, trust figures of merit may be utilized as a framework for benchmarking Trusted Part certifications.

Previous work conducted in Trust Metric development has focused on measures at the supplier level of abstraction [2] [3]. By quantifying the integrity of questionable hardware at the design level, one gains the granularity to address the trust concerns on a part-by-part basis. This work presents techniques to measure the integrity of hardware at the design level and arrives at a metric that can be used to gauge the design's trustworthiness. A questionable design is considered to be any design that has passed through an untrusted location within the supply chain (e.g. untrusted foundry.) This paper will review the parsing of the Design Integrity (DI) into five analyzable domains and discuss the measuring techniques for each. A novel method for quantifying the quality of the reference will be proposed and reviewed. The DI Analysis will then be applied to five test cases in order to evaluate and rank their respective integrities.

II. QUANTIFYING DESIGN INTEGRITY

Evaluating Design Integrity can provide valuable insight into vetting the trustworthiness of a design. The integrity of a design can be defined as the amount of deviation observed in a one-to-one mapping of the questionable design to a reference profile. These deviations can be caused by carelessly inserted faults, manufacturing flaws, or embedded Trojan circuitry. In essence, we are seeking an answer to the question, "*Does the design reliably operate the way that it was intended to without any anomalous behavior?*" Highest design integrity therefore consists of minimal deviation from the original specification. Lowest integrity is indicative of high deviation.

A. Parsing Design Integrity into Sub-Domains

In order to increase the insight into the design, we parse the design into five character sub-domains: Logical Equivalence, Signal Activity Rate, Structural Architecture, Functional Correctness, and Power Consumption. By evaluating each sub-domain, one acquires greater resolution into the design's characteristics from multiple viewpoints. Figure 1 illustrates conceptually the parsing of the design into the five sub-domain profiles. From here, both the expected and actual properties in each domain can be compared and the deviation between the two measured by a technique pertinent to the reference quality, amount of overdesign, and the domain being analyzed.

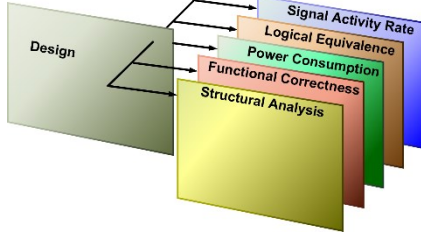


Figure 1 – Parsing Design into Sub-domain Profiles

Once all of the sub-domains are analyzed, their normalized deviation measurements can be aggregated together to arrive at the Design Integrity, DI , measure expressed as Equation (1) for the questionable design. Since each measurement is normalized, the different weights of each domain is accounted for by the domain weight factor, β_i , which takes the non-uniform nature of the aggregated domains into account. In this work, β_i was evaluated as uniform across all components.

$$DI = \text{Design Integrity} = \sum_{i=1}^n \beta_i T_i \quad (1)$$

where T_i = normalized domain measure technique and β_i = weighting for T_i . ($\beta_i = 1$)

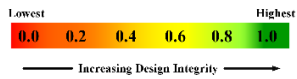


Figure 2 – Domain Specific Integrity Scale



Figure 3 – Aggregated Design Integrity Scale

Figure 2 displays the normalized DI scale used for a single sub-domain. Figure 3 represents the scale for the aggregation of all five sub-domains as determined by Equation (1).

B. Logical Equivalence Integrity Domain

The Logical Equivalence integrity, $LE_{integrity}$, assesses the degree to which the logic state points of the design in question map to the reference design. The process verifies the Boolean logic equivalence of a given design at the same or different levels of abstraction (e.g. RTL-to-Gate) by injecting test vectors to stimulate every logic state in the design. Comparison key points are defined to map a connection between the reference and questionable design. Each key point state is evaluated as logically

equivalent or non-equivalent. The equivalence check identifies the points between the two that are not equivalent, raising the concern for and identifying the location of the inserted error. Equation (2) and Equation (3) are expressions for the Equivalent Points and Total Comparison Points respectively.

$$\text{Points}_{EQ} = \sum_{i=1}^m b_i P_i \sigma_i \quad \text{where } b_i = \begin{cases} 0, P_i \text{ is } NEQ \\ 1, P_i \text{ is } EQ \end{cases} \quad (2)$$

$$\text{Points}_{COMPARED} = \sum_{i=1}^m P_i \sigma_i \quad \text{where } \text{Points}_{EQ} \subseteq m \quad (3)$$

$$LE_{integrity} = \frac{\text{Points}_{EQ}}{\text{Points}_{COMPARED}}, \quad 0 \leq LE_{integrity} \leq 1 \quad (4)$$

There are m Total Comparison Points. b_i is a binary value that evaluates if the comparison key point, P_i , was found to be equivalent. For cases where it was not equivalent, $b_i = 0$. For the case of an equivalent point, $b_i = 1$. The utilization factor of each P_i is represented as σ_i and gives more weight to higher utilized points and less weight to less utilized points. Finally, the metric $LE_{integrity}$ is expressed as Equation (4).

C. Signal Activity Rate Domain

The Signal Rate, SR , defines the number of times the evaluated element changes state over the duration of a given test scheme and is expressed in units of millions of transitions per second (Mtr/s) [4]. This effectively quantifies the activity rate or utilization of the analyzed signal and can be used to measure the deviation away from the expected activity of the design. Equation (5) determines the signal rate for element i where f_{CLK} is the clock frequency and σ_i is the utilization or toggle rate percentage of the element.

$$SR = f_{CLK} \sigma_i \quad (5)$$

The average signal rate for both the expected and actual designs is expressed in Equation (6). SR is divided into *data*, *input/output (IO)*, and *logic* components with a total signal quantity of a , b , c respectively for each. This can be determined for both the actual and expected and applied to Equation (7) for the deviation distance. Equation (8) is an expression for the normalized $SR_{integrity}$.

$$SR_{act,exp} = \frac{1}{a} \sum_{i=0}^a SR_{data} + \frac{1}{b} \sum_{i=0}^b SR_{IO} + \frac{1}{c} \sum_{i=0}^c SR_{logic} \quad (6)$$

$$\Delta SR_{dist} = |SR_{expected} - SR_{actual}| \quad (7)$$

$$SR_{integrity} = \frac{SR_{expected} - \Delta SR_{dist}}{SR_{expected}}, \quad 0 \leq SR_{integrity} \leq 1 \quad (8)$$

D. Power Consumption Domain

The Power Consumption domain, $P_{integrity}$, measures how closely the questionable design aligns to the original reference from a power perspective. Namely, for a given test scheme, how far does the power consumed by the actual design instantiation deviate away from the expected design? Equation (9) expresses

the expected and actual power consumptions, $P_{expected}$ and P_{actual} , at a power source point, i , in the design. Each power source point can be added together to arrive at a total power consumption for both the expected and actual power consumptions. The difference between the two can be represented as ΔP_{dist} and expressed as Equation (10). The final $P_{integrity}$ is determined by Equation (11).

$$P_{act,exp} = \left(V_i (I_{i,dynamic} + I_{i,static}) \right)_{act,exp} \quad (9)$$

$$\Delta P_{dist} = \left| \sum_{i=1}^n (P_{actual})_i - \sum_{i=1}^m (P_{expected})_i \right| \quad (10)$$

where n, m = total source points for the questionable and reference designs respectively

$$P_{integrity} = \frac{P_{expected} - \Delta P_{dist}}{P_{expected}}, \quad 0 \leq P_{integrity} \leq 1 \quad (11)$$

E. Functional Correctness Domain

The Functional Integrity, $F_{integrity}$, is evaluated by observing the number of errors that occur, $\epsilon_{observed}$, for a given verification test scheme and can be expressed as Equation (12). TP_{total} is the total verification test points used for verifying the design functionality. $\epsilon_{observed}$ is the number of error cases accumulated from the test scheme.

$$F_{integrity} = \frac{TP_{total} - \epsilon_{observed}}{TP_{total}}, \quad 0 \leq F_{integrity} \leq 1 \quad (12)$$

The verification test scheme is designed to stimulate both the original reference and actual design so the functionality of both designs can be compared. Test schemes range from exhaustive testing to ones that only provide corner and basic functional coverage. For every test where the actual design does not match the expected result, an error is observed. $F_{integrity}$ can then be expressed as the ratio of successful tests (i.e. Expected Result equals Actual Result) to the total tests made.

F. Structural Analysis Domain

The Structural Analysis looks at the architecture components of the design that are generated once the design has been synthesized into a gate level netlist. For an FPGA, when the synthesis process is executed, the design Nets and Leaf Cells are represented hierarchically as subcomponent architectures. As such, these become the points of comparison for identifying any deviation from the expected structure. Equation (13) and (14) determine the number of extra or removed Nets and Leaf Cells respectively for an evaluated architecture component i .

$$S_{\Delta X_i} = |Nets_{expected} - Nets_{actual}| \quad (13)$$

$$S_{\Delta Y_i} = |Cells_{expected} - Cells_{actual}| \quad (14)$$

The modified Nets and Cells can then be represented as a ratio against the total Nets and Cells to arrive at the Structural Integrity, $S_{integrity}$, expressed in Equation (16). In order to maintain the resolution of the modified circuits from getting washed out in a large design, only the architectures that show a modification to the Nets or Leaf Cells are considered; therefore $S_{\Delta X_i} \neq 0$ and $S_{\Delta Y_i} \neq 0$.

$$\Delta S = \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^n \left[\frac{S_{\Delta X_i}}{X_{i,expected}} \right] + \frac{1}{m} \sum_{i=1}^m \left[\frac{S_{\Delta Y_i}}{Y_{i,expected}} \right] \right) \quad (15)$$

$$S_{integrity} = 1 - \Delta S \quad (16)$$

where n, m = number of modified architectures evaluated for Nets, Leaf Cells respectively ($S_{\Delta X_i} \neq 0$ and $S_{\Delta Y_i} \neq 0$)

III. TEST CASE TO EVALUATE DESIGN INTEGRITY

A. Floating Point Adder System

Several test cases were setup in order to demonstrate the usefulness of the DI metric. Figure 4 shows a block diagram of the test system, comprised of two Fixed Point Converters, a Floating Point Adder, and an Output Buffer. The system allows two 12-bit fixed point inputs to be converted into single precision IEEE 754 Standard Floating Point Format. The two values are then added together and the result observable at the system output.

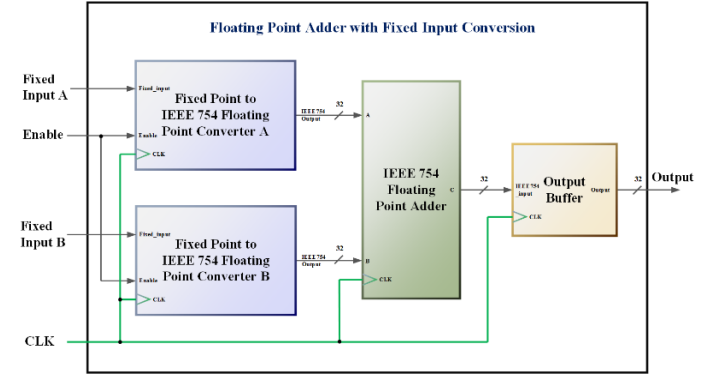


Figure 4 – Test Case Block Diagram

The system was corrupted with the addition of several errors ranging from Stuck-At faults to well-hidden malicious Trojans. This created a spectrum of errors with varying payloads (i.e. damage capability) to mimic adversarial tampering. Table 1 presents details of each error as well as the insertion location and activation mechanism (i.e. trigger.)

Test Article	Error Location	Error Trigger	Description
No Error TA	None	None	No malicious circuitry added to design
Error 1 TA	Output Buffer	Time Bomb with Counter	Denial of Service attack launched once pre-set time count is met
Error 2 TA	Output Buffer	Counter Trigger	Slows down performance through counter delays
Error 3 TA	Top Module	Siphon Enable	Data is siphoned off unmonitored port when requested
Error 4 TA	Fixed to IEEE754 Conversion	No Trigger	30 th bit of converter output stuck at logic HIGH

Table 1 – Description of Errors Inserted into Test System

Table 2 presents the results of the analysis applied to each of the test article designs. Each of the domains are evaluated on a [0, 1] scaling and are marked with a color indicative of the DI scale shown in Figure 2. The Design Integrity is consistent with the scale of Figure 3. One can see that the integrity for the No Error TA was highest followed by the Error 4 TA. Errors 1 and 2 TAs were quantified with the lowest integrities. Based on the analysis, the DI metric shows measurable differentiation between all five

of the test cases and lends itself to the ranking of each test article in the order of highest to lowest trust.

Test Article	P _{integrity}	F _{integrity}	SR _{integrity}	S _{integrity}	LE _{integrity}	Design Integrity
No Error TA	1.0000	1.0000	1.0000	1.0000	1.0000	5.00
Error 1 TA	0.7647	0.0034	0.8555	0.9424	0.7083	3.27
Error 2 TA	0.7059	0.1330	0.7365	0.6102	0.7648	2.95
Error 3 TA	0.7059	1.0000	0.9956	0.8803	0.8016	4.38
Error 4 TA	0.9412	0.4993	0.9938	0.9704	0.9949	4.40

Table 2 – Design Integrity Results for Test System

B. Quantifying the Reference Quality

One question that intuitively rises when investigating the integrity analytics revolves around the quality of reference being utilized in the analysis. As such, formulating a metric for quantifying the reference quality, R_Q , and correlating it to the obtained DI metric allows one to place higher or lower confidence in the DI measures. It also lends itself to being used for comparing different reference types and ranking one against another in terms of usefulness. Reference quality is determined by Equation (17) where n is the number of integrity domains the reference can evaluate and N the total possible domains to evaluate.

$$R_Q = \frac{n}{N}, \text{ where } 0 \leq R_Q \leq 1 \text{ and } n \leq N \quad (17)$$

Table 3 shows five different references that were used in the DI analysis and how they were scored. R_Q can be used in conjunction with the DI metric to arrive at a final design Trust Measure expressed in Equation (18) that is indicative of the confidence one can have in the insights afforded by the DI metric. Equation (19) represents the normalized DI used to bring any number of domains evaluated into the [0, 1] scale system.

Reference No.	Analyzable Domains					Reference Quality (R_Q)	Description and Format
	P	F	SR	LE	S		
Reference 1	X	X	X	X	X	5.00	Synthesizable Behavioral Design (VHDL)
Reference 2	X	X	X	X		2.00	Datasheet Specification (MS Word)
Reference 3	X					1.00	Executable Specification (MATLAB)
Reference 4	X	X	X	X		3.00	Datasheet with Executable Specification (MATLAB/MS Word)
Reference 5	X	X	X	X	X	4.00	Synthesized Netlist (Verilog)

Table 3 – Description of References

$$\text{Trust Measure} = TM = DI_{norm} \cdot R_Q \quad (18)$$

$$DI_{norm} = \frac{DI}{n} \quad (19)$$

Table 4 revisits the DI metrics presented in Table 2 and shows how the TA cases would be evaluated with different references. Reference 1 was the highest quality because it applied to all five domains of the DI analysis. Reference 3 was the lowest quality lending itself to be utilized in only one domain. The impact of reference quality on quantifying design integrity is highlighted with the Error 3 TA example. DI_{norm} was measured as 0.88 for Reference 1, but measured as 1.00 by Reference 3. This is because Reference 3 does not afford the level of observability that Reference 1 does into the design to track the deviations caused by the error. Based on this information, one could be led to believe that Error 3 TA was of highest trust. The Trust Measure however accounts for the poor reference quality and adjusts the scoring to

0.20 which is significantly lower than the Reference 1 Trust Measure of 0.88.

Test Article	N	Reference 1 ($R_Q = 1$)					Reference 2 ($R_Q = 2/5$)					Reference 3 ($R_Q = 1/5$)				
		n	R_Q	DI	DI_{norm}	TM	n	R_Q	DI	DI_{norm}	TM	n	R_Q	DI	DI_{norm}	TM
No Error TA	5	5	1	5.00	1.00	1.00	2	0.4	2.00	1.00	0.40	1	0.2	1.00	1.00	0.20
Error 1 TA	5	5	1	3.27	0.65	0.65	2	0.4	1.71	0.85	0.34	1	0.2	0.00	0.00	0.00
Error 2 TA	5	5	1	2.95	0.59	0.59	2	0.4	1.32	0.66	0.26	1	0.2	0.13	0.13	0.03
Error 3 TA	5	5	1	4.38	0.88	0.88	2	0.4	1.59	0.79	0.32	1	0.2	1.00	1.00	0.20
Error 4 TA	5	5	1	4.40	0.88	0.88	2	0.4	1.91	0.96	0.38	1	0.2	0.50	0.50	0.10

Test Article	N	Reference 4 ($R_Q = 3/5$)					Reference 5 ($R_Q = 4/5$)					Trust Scaling	
		n	R_Q	DI	DI_{norm}	TM	n	R_Q	DI	DI_{norm}	TM		
No Error TA	5	3	0.6	3.00	1.00	0.60	4	0.8	4.00	1.00	0.80	0.80 - 0.99	Highest Trust
Error 1 TA	5	3	0.6	1.71	0.57	0.34	4	0.8	3.27	0.82	0.65	0.60 - 0.79	
Error 2 TA	5	3	0.6	1.45	0.48	0.29	4	0.8	2.82	0.70	0.56	0.40 - 0.59	
Error 3 TA	5	3	0.6	2.59	0.86	0.52	4	0.8	3.38	0.85	0.68	0.20 - 0.39	
Error 4 TA	5	3	0.6	2.41	0.80	0.48	4	0.8	3.90	0.98	0.78	0.00 - 0.19	Lowest Trust

Table 4 – Comparison of Different References Types

IV. CONCLUSION AND FUTURE WORK

This paper proposed several techniques for evaluating a design's integrity by looking at five different characteristic domains of the design and then aggregating their measured deviations from expected characteristics together to arrive at a single value DI metric. A novel method for quantifying references was also presented that considers the quality of the reference being used in the DI Analysis. Quality of the reference is a way to capture amount of overdesign into the DI metric. A final Trust Measure indicative of the design's integrity and the confidence provided by the reference quality was achieved. The major takeaway from this work is that one can now quantifiably indicate that a design has poorer or higher integrity and measurably present how far it has deviated away from the expected characteristics. By establishing reference quality metrics, the quality of the DI value obtained from different references can be compared and different references ranked according to their utility.

The development of trust metrics is an iterative process with future iterations improving on and adding to the previous ones. New domains for characterizing the design need to be explored to increase the observability into hardware. The weighting factor, β_i , of each domain also remains to be determined to address the normalization approach taken in each sub-domain.

V. ACKNOWLEDGEMENTS

The work reviewed in this paper was conducted with sponsorship from AFRL and the Dayton Area Graduate Studies Institute (DAGSI.)

VI. REFERENCES

- [1] A. Thierier and A. Castillo, "Projecting the Growth and Economic Impact of the Internet of Things," Mercatus Center - George Mason University, 15 June 2015. [Online]. [Accessed 23 September 2015].
- [2] S. Moein and F. Gebali, "A Formal Methodology for Quantifying Overt Hardware Attacks," in *Advances in Information Science and Computer Engineering*, Dubai, 2015.
- [3] D. Pentrack, L. Neal, J. Lloyd and A. Gahoonia, "Quantifying System Trust and Microelectronics Integrity," in *GOMAC*, 2015.
- [4] Xilinx Inc., "Vivado Design Suite - UG907 Power Analysis and Optimization," 25 July 2012.