GOMACTech08

To: All GOMACTech Participants

From: Gerry Borsuk, Conference Chair

Subject: GOMACTech 2008 Proceedings -- Distribution Clarification

Date: June 1th, 2008

1. Distribution Statements are used in marking technical data to denote the extent of its availability for secondary distribution, release, and disclosure without additional approvals or authorizations by the originator or controlling office. (DoDD 5230.24)

2. Be advised that the GOMACTech 2008 Proceedings CD-ROM as a compilation is Distribution X.

3. All unmarked component papers are to be controlled and handled in accordance with Distribution X.

3. The markings on the following two papers are changed to Distribution X:

> Title: "Family of UGS Demonstration" by Army Research Laboratory (ARL)
> Distribution C is changed to Distribution X
>
> Title: "Defining MAC1 Components Through a Top Down Approach" from SAIC
> FOUO is changed to Distribution X

4. Only those papers marked "Approved for Public Release" have no distribution constraints.

5. Please keep this letter with your copy of the GOMACTech 2008 Proceedings on CD-ROM.

G W Borsuk

Gerry Borsuk, GOMACTech 2008 Conference Chair

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAR 2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE **Metrics for TRUST in Integrated Circuits**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Johns Hopkins University,Applied Physics Laboratory,11100 Johns Hopkins Road,Laurel,MD,20723-6099**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

See also ADM202438. Presesnted at GOMACTech-08, Microsensor Technologies: Enabling Information on Demand, 17-20 Mar 2008, Las Vegas, NV

14. ABSTRACT

In this paper we report on metrics approaches adapted for the DARPA TRUST in ICs program. A metrics approach initially focused on detection of malicious alterations in integrated circuit die has been adapted for use on FPGA bitstreams and the ASIC design process. We also discuss metrics for techniques focused on prevention of malicious alterations.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**5**	

Metrics for TRUST in Integrated Circuits

Daniel P. Wilt, Richard C. Meitzler, and John P. DeVale
Milton S. Eisenhower Research Center
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD 20723-6099
Daniel.Wilt@jhuapl.edu

Abstract: *In this paper we report on metrics approaches adapted for the DARPA TRUST in ICs program. A metrics approach initially focused on detection of malicious alterations in integrated circuit die has been adapted for use on FPGA bitstreams and the ASIC design process. We also discuss metrics for techniques focused on prevention of malicious alterations.*

Keywords: trust; metrics; Trojan; detection

Introduction

In the Defense Science Board report, "DSB Task Force on High Performance Microchip Supply" [1] several issues associated with the future acquisition of high performance integrated circuits (ICs) are raised. Among the issues identified, a particularly serious one is the potential for insertion of malicious circuitry into ICs by untrusted parties in the design and fabrication process. This hazard stems from the existing, and increasing, need to use design, fabrication, and packaging facilities and foundries for the acquisition of DoD-required ICs which are outside the span of U.S. Government control. Of most serious concern are those facilities which are foreign-owned or foreign-operated. These uncontrolled facilities create many new opportunities for potential adversaries to manipulate the content and functionality of ICs whose performance is critical to current and future U.S. military capabilities.

A *trusted* integrated circuit is one in which, through the application of appropriate IC design, fabrication, and measurement technology, a degree of confidence has been developed that no malicious circuit changes have occurred. The range of potential alterations is quite broad and no single design, fabrication, or measurement technology alone is anticipated to provide adequate confidence that a given integrated circuit design is unchanged.

In a previous paper [2], we considered the application of techniques for the determination of trust in an integrated circuit in a very general sense. In that paper, we proposed and developed a metrics methodology based upon measurement of the probability of detection of a malicious circuit insertion in an IC, P_d, and the corresponding probability of false alarm, P_{fa}. With these definitions, the performance of a Trust-related technology is determined by its receiver operating characteristic (ROC) curve relating P_d and P_{fa} as a function of chosen parameters including, for example, decision thresholds. We further developed the relationship between the IC-level probability of detection, P_d, and probability of false alarm, P_{fa}, and the probabilities for detection $P^t{}_d$ and false alarm $P^t{}_{fa}$ for single changed elements such as individual transistors or wires. We also considered techniques operating at a resolution intermediate between transistor/wire and full IC, and developed mathematical techniques to relate these measures to the IC-level P_d and P_{fa} and transistor/wire level $P^t{}_d$ and $P^t{}_{fa}$.

In that paper we further discussed the issue of assessing P_d and P_{fa} when results from different techniques are combined together to make an aggregate decision regarding an integrated circuit's trustworthiness. One issue with combined technical approaches is that the results of the different techniques may correlate, in which case a simpleminded estimate of the combined P_d assuming independent measurements by each technique could be overly optimistic. In this situation, the use of a Bayesian Network (BN) approach allows estimation of the dependent relationships between measurements and proper estimation of the combined P_d and P_{fa}.

This approach to metrics for trust is quite different from a more traditional security approach based upon attack tree construction, probabilistic threat assessment, and risk mitigation [3]. The traditional approach is not considered to be appropriate for Trust, since we assume that adversaries already have adequate access to the design and fabrication chain to implement circuit changes. The question is not whether an adversary is able to make changes, which is a given. The question of interest is, if an adversary chooses to make changes, can they be detected? As an alternative possibility, is there a way to prevent an adversary's inserted changes from performing their intended function?

DARPA's TRUST in ICs Program

In this paper, we discuss the application of the methodology presented in the previous paper to the DARPA "TRUST in ICs" program. This program is a broad and far-reaching attempt to develop technology which would insure that an IC whose design and fabrication involves untrusted parties contains the intended circuitry and functionality, and nothing else.

The TRUST in ICs program is not only interested in the problem of finding changes implemented in the *fabrication* processes of an ASIC, it is also interested in the problem of finding changes inserted in the *design* process for the ASIC prior to submission of the design to a foundry. Such changes could potentially occur as a result of inclusion of IP blocks from untrusted sources or the use of design tools obtained from untrusted sources.

Additionally, the TRUST in ICs program is interested in finding changes inserted into the design and loading processes for an *FPGA configuration bitstream*. FPGAs are increasingly important in defense applications due to their cost advantage for low-volume applications, high performance, short design cycle, and flexibility. The FPGA configuration bitstream experiences similar threats to an ASIC design including untrusted IP blocks and untrusted tools.

Metrics for TRUST in ICs

In a typical situation of interest to the TRUST in ICs program, at one point in the design/fabrication cycle, the design is trusted. That point may be as far upstream as the initial specification or RTL coding of the design, or it may be far downstream such as when the GDS-II design data is submitted to a foundry for fabrication. The design flow of interest is diagrammed in Fig. 1 below. The trusted design then passes through an untrusted step (or steps) in which a potential adversary has access to it. After the untrusted step is performed, the design may have been intentionally changed in order to implement a desired malicious functionality in the end product.

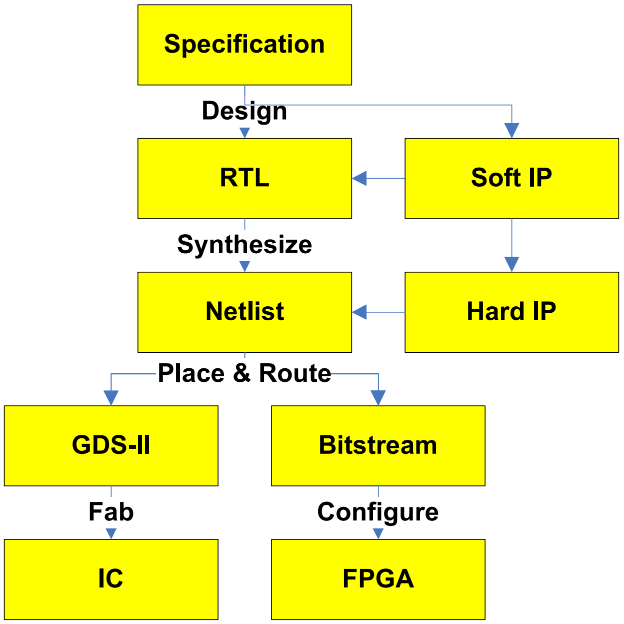

Figure 1. Design flow

After the design passes through the untrusted step, it is again available to a trusted party to make a measurement and compare with the trusted design prior to the untrusted step.

Of course, the design comparison in question is typically between two very different representations, which potentially makes the comparison very difficult. For example, the trusted input may be a GDS-II CAD file submitted to an untrusted foundry, and the output to be compared would thus be the final, fabricated integrated circuit die. The required comparison could involve determining the actual layered structure of the fabricated IC through physical reverse engineering techniques, including the implanted/diffused doped structures, contacts, poly, vias, and all wiring levels, and then performing a comparison between this extracted data and the intended design. Furthermore, the fabricated IC may include intentional but benign modifications performed by the foundry such as *fill* and *cheese* (changes to the metal wiring structures to improve yields in chemo-mechanical polishing processes). The comparison process needs to be robust enough to ignore these expected differences and focus instead on those changes which are unexpected and potentially malicious.

It is important to understand that the potential adversary may choose not to change a given design when it passes through his hands, if for example he judges the risk and consequences of exposure outweigh the potential benefits. Thus, it is not possible to establish trust on the basis of past good performance. It must instead be established on each design that passes through the untrusted process on an individual basis, and established based upon data that is known and measured only by trusted parties.

Transistor Level Metrics

For the TRUST in ICs program, we have chosen to focus the metrics on the ability to detect a change at the smallest circuit element level, i.e. an individual transistor or net. The metric goals for the program are stated in terms of P^t_d and P^t_{fa}, probabilities of detection and false alarm for individual circuit elements in the IC.

For a comparison between a GDS-II CAD file and an actual IC, these transistor level metrics are a natural representation of what is being measured and compared in the reverse engineering process, and it is straightforward to calculate the desired transistor-level metrics from actual measured data. Similarly, for a comparison of a trusted netlist to a GDS-II CAD file, the changed elements of the GDS-II are easily interpreted as transistors or nets. However, for several other situations, the detected changes are not so easily interpreted, and some discussion is necessary. In particular we consider three cases: (a) comparison of a trusted RTL formulation to an untrusted netlist, (b) comparison of a trusted netlist to an untrusted FPGA configuration bitstream, and (c) comparison of a trusted hard-macro IP specification to an untrusted GDS-II layout.

For the comparison of a trusted RTL to untrusted netlist, the natural form to report changes would be elements of the untrusted netlist which are inconsistent with the source RTL. This could include insertions, deletions, and changes. Conceptually, this is similar to formal equivalence checking but with the added twist of locating intentional changes, potentially intentionally camouflaged as well, rather than errors. In terms of counting transistors, the identified netlist elements can be referenced to a placed and routed GDS-II file which has a definite transistor and net count. There is of course some ambiguity since the place and route process typically includes the addition of supporting circuitry such as buffers and clock trees to achieve acceptable performance.

For the comparison of a trusted netlist to an FPGA bitstream, the natural form to report changes would be in terms of the configured primitive elements of the FPGA. Those primitive elements are vendor-specific and product-specific. From the standpoint of demonstrating acceptable false alarm rates, it is desirable to report changes in terms of the smallest possible primitive elements, such as lookup-tables, flip-flops, multiplexers, and gates. Based upon the logical function of each primitive, a transistor count may be assigned in order to calculate transistor-based metrics.

For the comparison of a trusted specification to an untrusted hard-macro IP block, we have chosen to represent discrepancies in the form of text descriptions of the changed functionality. In this case, a description of the changed functionality in the GDS-II hard-macro representation would require a full understanding of the implementation, while a textual description of the changed functionality seems a more efficient and feasible task. For transistor counts, knowledge of the changes in the GDS-II suffices to assign exact transistor counts to the changed functions.

Statistical Significance

The DARPA TRUST in ICs program has specific target go/no-go milestones for each phase. The Phase I go/no-go milestones are shown in Table 1.

Table 1. TRUST in ICs Phase I go/no-go milestones

	Untrusted FAB	Untrusted Design ASIC	Untrusted Design FPGA
P^t_d	90.0%	80.0%	90.0%
P^t_{fa}	10^{-3}	10^{-3}	10^{-3}
N	10^5	10^5	10^5
Time	480H	480H	480H

In this table, N represents the total number of transistors in the tested IC, and *Time* represents the total amount of time in hours (human and machine) to determine trust in a given design. The P^t_d, P^t_{fa}, and the resolution of the detection technique r (in transistors) constrain the size of test articles and the number of inserted changes. In particular, to establish P^t_{fa} with acceptable confidence on a single test article places a constraint on the minimum size in transistors for the test article, based upon the following relationship:

$$P^t_{fa}\Big|_{upper} = BETAINV\left(C, \frac{n}{r}+1, \frac{N-n}{r}+1\right)$$

Where $P^t_{fa}|_{upper}$ is an upper bound on P^t_{fa} with confidence C, *BETAINV* is the quantile function for the beta statistical distribution (available in Excel), n is the total number of transistor false alarms, and N is the total number of transistors. From this equation, for a resolution of $r = 1$ transistor, the minimum test article size to establish $P^t_{fa} = 10^{-3}$ with 90% confidence is $N = 2302$ transistors assuming $n = 0$ false alarms. Conversely, for a test article meeting the required size of $N = 10^5$ evaluated transistors, with a resolution $r = 1$ transistor up to 87 transistor false alarms could occur and still meet $P^t_{fa} < 10^{-3}$ with 90% confidence. Finally, for a test article of $N = 10^5$ transistors and zero transistor false alarms, the resolution r must be less than 43 transistors to meet $P^t_{fa} < 10^{-3}$ with 90% confidence. Based upon this latter calculation, we conclude that any technique which can only resolve trust on a full-die level (e.g. based upon a full-die signature) would require many test articles in order to establish the required P^t_{fa}.

Another relationship holds for P^t_d. In this case,

$$P^t_d\Big|_{lower} = BETAINV\left(1-C, m+1, M-m+1\right)$$

Where $P^t_d|_{lower}$ is a lower bound on P^t_d with confidence C, m is the number of detected Trojan transistors, and M is the total number of Trojan transistors. From this relationship, in order to establish $P^t_d = 90\%$ at 90% confidence on a single test article, we must have at least $M = 21$ Trojan transistors to detect, assuming all are detected ($m = M$). For a much larger number of changed transistors, e.g. $M = 1000$, to establish $P^t_d = 90\%$ at 90% confidence requires $m = 913$ detected transistors. Asymptotically, $m = 0.9M$.

Prevention Techniques

While the main focus of the TRUST in ICs program is on detection of changes in an untrusted design/fabrication flow, there is a prevention scenario worth considering. In this case, through modification of the design or design process, malicious changes might be prevented from exercising their functionality. An example of this might be a self-repairing circuit that monitors and responds to its own unexpected behavior. In this prevention scenario, we can consider measuring P_p, the probability of preventing the malicious functionality from being exercised. Similarly, there is a corresponding P_{fp}, or the probability of false prevention. In this context, P_{fp} could be interpreted as the probability that an acceptable design cannot be successfully implemented with the new process, or fails to operate properly (including unacceptable performance).

It is important to consider that TRUST presumes a very capable adversary, one who potentially has the resources of a nation-state. From that viewpoint, merely obscuring the design in a manner which might defeat a hacker or criminal should be considered inadequate for TRUST purposes. For TRUST, a prevention scenario must be firmly based upon the limitations of known technology as a nation-state would experience them – for example, the level of difficulty in the factoring of large composite numbers which is the basis of modern cryptography.

Conclusion

In this paper, we have described the process of implementing metrics for the DARPA TRUST in ICs program. We have focused the metrics for this program on the probability of detection of maliciously inserted transistor changes, P^t_d, and the probability of false alarm, P^t_{fa}. These metrics can be applied to a number of potential TRUST scenarios, including untrusted fabrication facilities, untrusted design tools, and untrusted COTS FPGAs. Requirements on test article size and number of inserted changes in order to establish P^t_d and P^t_{fa} meeting the Phase I go/no-go goals were presented. The required test article sizes and number of inserted changes are easily realizable for Phase I tests. Finally, the opportunities for prevention techniques were discussed.

Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency, Microsystems Technology Office (DARPA-MTO) under contracts MDA972-01-D-0005 and HR0011-06-D-0003.

References

1. "DSB Task Force on High Performance Microchip Supply," Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, D.C. 20301-3140, issued February 2005.
2. Wilt, D. P. and Meitzler, R. C., "Metrics for Trusted Integrated Circuits" *GOMACTech 2007 Proceedings*, Orlando, FL, March 19-22, 2007, paper 11.2, pp. 133-135.
3. Michael Howard and David LeBlanc, *Writing Secure Code, Second Edition*, Redmond, WA, Microsoft Press, 2003.