



**IEEE P3164: SECURITY ANNOTATION FOR
ELECTRONIC DESIGN INTEGRATION**

**ASSET IDENTIFICATION
FOR ELECTRONIC DESIGN IP**

Authored by

IEEE P3164 Security Annotation for Electronic Design Integration (SA-EDI)
Working Group

TRADEMARKS AND DISCLAIMERS

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

The ideas and proposals in this specification are the respective author's views and do not represent the views of the affiliated organization.

ACKNOWLEDGMENTS

Special thanks are given to the following authors of this paper:

Brent Sherman, Chair, *Intel*
Kamran Haqqani, Vice Chair, *Accenture*
Adam Sherer, Secretary, *Cadence*

Sohrab Aftabjehani, *Intel*
Jessy Ayala, *UC Irvine*
Debojyoti Bhattacharya, *Arm*
Mike Borza, *Synopsys*
Akim Layachi Daineche, *Arm*
Trenton Grale, *Accenture*
Miltos Grammatikakis, *Hellenic Mediterranean University*
John Hallman, *Siemens*
Kathy Hayashi, *Qualcomm*
Pavani Jella, *Silicon Assurance*
Wayne Kohler, *Perforce Software*
Amit Kumar, *Microsoft*

Yann Le Floch, Self
Wenzhen Li, Self
Jean-Philippe Martin, *Intel*
Mehdi Mohtashemi, *Intel*
Jean-François Mousinho, *Synopsys*
Prasad Nandipati, *Qualcomm*
Anders Nordstrom, *Cyccuity Inc.*
James Pangburn, *Cadence Design Systems*
Rafael Santos, *Arm*
Benjamin Tan, *University of Calgary*
Vivek Vedula, *Arm*

The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.

All rights reserved. 5 April 2024. Printed in the United States of America.

PDF: STDVA26891 979-8-8557-0661-1

IEEE is a registered trademark in the U. S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated. All other trademarks are the property of the respective trademark owners.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system, or otherwise, without the prior written permission of the publisher.

Find IEEE standards and standards-related product listings at: <http://standards.ieee.org>.

NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA DOCUMENTS

This IEEE Standards Association (“IEEE SA”) publication (“Work”) is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the activity that produced this Work. IEEE and the IEEE P3164 expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE P3164 disclaim any and all conditions relating to: results; and workmanlike effort. This document is supplied “AS IS” and “WITH ALL FAULTS.”

Although the IEEE P3164 members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE SA OR IEEE P3164 MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patent claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder.

This Work is published with the understanding that IEEE and the IEEE P3164 members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

TABLE OF CONTENTS

ABSTRACT.....	5
1. INTRODUCTION	5
2. GENERAL.....	7
3. ASSET IDENTIFICATION	7
3.1. CONCEPTUAL AND STRUCTURAL ANALYSIS	8
3.1.1. CONCEPTUAL ANALYSIS.....	9
3.1.2. STRUCTURAL ANALYSIS	10
3.2. EXAMPLES.....	10
3.2.1. SIMPLE GPIO PAD	10
3.2.2. GAUSSIAN NOISE GENERATOR	12
3.2.3. AES ENGINE	14
3.2.4. SRAM CONTROLLER.....	16
4. POINTS OF INFLUENCE AND OBSERVATION	19
4.1. GENERIC CPU CORE	19
4.1.1. GENERAL.....	19
4.1.2. POINTS OF INFLUENCE AND OBSERVATION	21
5. SUMMARY	23
6. REFERENCES.....	24
APPENDIX A ABBREVIATIONS, ACRONYMS, AND DEFINITIONS	25

ASSET IDENTIFICATION FOR ELECTRONIC DESIGN IP

ABSTRACT

The Accellera Security Annotation for Electronic Design Integration (SA-EDI) standard [1]¹ provides a framework for producing security assurance collateral for an IP. The root of the standard and its workflow is the identification of assets for a given IP. Once the assets are identified, the corresponding threats and attack surfaces can be determined to help an Integrator address risks in their integrated circuits (ICs). Therefore, if an asset was mistakenly identified, either a false positive or a false negative, the rest of the collateral would become invalid. Unfortunately, the standard provides little guidance on identifying assets and avoiding false positives/negatives. This paper proposes two methodologies for asset identification within an IP using conceptual and structural analysis (CSA) and points of influence and observation (PIO).

1. INTRODUCTION

The Security Annotation for Electronic Design Integration (SA-EDI) standard builds on the identification of assets within the IP. The Element and Attack Point Security Objective (APSO) objects are built using information from the Asset Definition objects. If the information in the Asset Definition objects is not accurate, then the collateral produced from it becomes inaccurate, thus leading to potential exploits in the integrator's integrated circuit (IC). Therefore, assets must be accurately identified and classified accordingly. Unfortunately, SA-EDI does not provide enough guidance in this domain to reduce false positives and false negatives. In addition, contextual IC information such as security requirements, use cases, and so on are not known to the IP developer, making the creation of SA-EDI collateral more difficult. This contextual information is only known well after the IP has been developed and delivered to the integrator.

¹ Numbers in brackets correspond to sources listed in Section 6.

This document proposes a new methodology and classification guidance to help identify assets in an IP. It walks through the methodology using several example IPs that identify and associate high-level assets to security objectives, resulting in the definition of structural assets in the Register-Transfer Level (RTL). These IPs are listed as follows:

- **Simple GPIO Pad:** Single GPIO with Direction Select.
- **Gaussian Noise Generator (GNG)** [2]: A GNG generates random noise that has a standard normal distribution and constant power spectral density. The core uses a 64-bit combined Tausworthe generator (LFSR and XOR) and further processes to transform random noise distribution to white Gaussian noise (GN) [with blocks split, mask, leading zero detector (LZD), swizzle, and address]. A polynomial approximation with specific coefficients is used. The generator also relies on an initial seed.
- **Advanced Encryption Standard (AES) Engine:** A high-level AES engine with separated data in/out paths. Configuration registers set the mode, encrypt/decrypt operation, key size, and start. Read-only status registers report errors, while the data out signal provides encrypted/decrypted output when it is available.
- **SRAM Controller** [5]: A simple static random-access memory (SRAM) controller that supports a sleep mode and memory built-in self-test (MBIST).
- **Generic CPU Core:** A basic core pipeline and execution unit with instruction and data caches.

Section 2 details the SA-EDI standard workflow at a higher level, examining how this methodology can be used to create security assurance collateral, mainly focusing on the Asset Definition objects. Section 3 details a new methodology called *conceptual and structural analysis* (CSA) to identify IP assets to create the Asset Definition objects. It also demonstrates the approach by applying it to several IP examples.

NOTE—Only the GNG and SRAM examples reference the actual code since the RTL is available in their respective repository.

Section 4 introduces another method specifically for complex IPs or IPs with multiple shared resources. This method is called *points of influence and observation* (PIO) and can be used in addition to CSA or as an alternative. Section 5 summarizes both approaches and use cases.

Lastly, this standalone document is intended to be used as supplemental guidance to v1.0 of the Accellera SA-EDI standard. Future versions of the standard may or may not impact the content of this document.

2. GENERAL

The SA-EDI standard focuses on creating security assurance collateral for proper integration of an IP. The standard defines four objects, as a progression, which are included in the IP bundle the IP provider creates. These objects and their associations are shown in Figure 5 of the SA-EDI standard [1]. At the root of the object creations are the Database and Asset Definition objects. These objects are the starting points for creating the collateral.

The Database object points to a security weakness knowledge base (e.g., MITRE CWE [6]), as defined in SA-EDI, which contains a database of known weaknesses. This object is optional, however, highly recommended, especially when using an industry-available database, such as CWE. The Asset Definition object points to assets within the IP. These assets may be a module, register, buffer, array, or any material defined by the RTL. Both objects, along with the RTL, are used to produce the Element objects as shown in Figure 3 of SA-EDI [1]. The Element object is used to associate IP top-level ports (i.e., inputs and outputs) to an asset. These ports are the attack surface for a threat to a security objective an asset may have. For example, a confidentiality security objective may be violated by a port if it can be used to observe an asset at the integration level of the IP. This information (i.e., assets, ports, security objectives, threats, etc.) is captured in Attack Surface Security Objective (APSO) objects, which is shown in Figure 3 of SA-EDI [1]. The APSO objects are the end results that are used by the integrator to help generate the threat model of the IC, as shown in Figure 4 of SA-EDI [1].

The Asset Definition objects form the foundation for creating the APSO objects. If they are not properly constructed, for example, if they are questionable in validity, then the result will be invalid APSO objects. Determining the validity of an Asset Definition object is subjective, meaning that what one developer deems as an asset may or may not be what another developer considers as an asset. This becomes the root of the problem; how can one objectively identify, with justification, what an IP asset is? This paper proposes a dual-approach to help increase the confidence that an asset has been properly identified, thus resulting in valid APSO objects for the integrator to consume. Both methodologies require the use of an architectural block diagram. As a disclaimer, this document does not eliminate subjectivity. It merely provides two structured approaches to help the IP provider apply a security mindset.

3. ASSET IDENTIFICATION

Identifying assets and their properties is a way to help discover risks associated with IP integration. Many use this approach as the first step in threat modeling, especially when dealing with a large interface or attack surface.

As with most approaches, the first step is often the critical piece since the rest of the process builds from it. Therefore, any inaccuracies, such as false positives or negatives, may produce erroneous results. In security, this can lead to exploitation. As an example, if the threat modeling process missed an asset (i.e., a false negative), then a threat could remain unidentified resulting in a missing mitigation. To help address this concern, the following two approaches are proposed: CSA and PIO.

The CSA approach focuses on security objectives to identify conceptual assets. These conceptual assets are used to identify their RTL representation, such as registers, modules, gates, and so on, which are the structural assets. These structural assets create the Asset Definition objects in the SA-EDI standard. The PIO method starts by looking at the interface of the IP to identify conceptual assets. Once identified, the focus becomes points of influence and observation on these assets. These are points through which an asset may be compromised. If a security objective can be compromised, then the RTL representation of these points would be a structural asset and thus require an Asset Definition object.

3.1. CONCEPTUAL AND STRUCTURAL ANALYSIS

The concept behind CSA is to use conceptual assets to identify structural assets that will be used to define the Asset Definition objects of the SA-EDI standard. A conceptual asset is a high-level asset associated with the use-case flows of the IP, such as data and system state, which involves a security objective such as confidentiality, integrity, or availability (CIA). An example of a conceptual asset is data that a user wants to identify as a secret. This data would, at least, require confidentiality as a security objective, thus making it an asset.

A structural asset is RTL material that physically supports a conceptual asset. Examples of structural assets may be registers, modules, flops/latches, gates, and so on. Using the plaintext data as the conceptual asset, a structural asset would be RTL material that supports its data flow. An example may be a buffer that temporarily stores the data as it is entered into the IP. Another example may be registers that contain details about this data, which should also be kept secret. These examples may be considered structural assets, thus requiring SA-EDI Asset Definition objects.

It is worth noting that the proposed methodology does not claim to eliminate ambiguity or subjectivity when determining what is an asset. Instead, the claim is that CSA is one of many approaches that an IP developer can use to help generate SA-EDI collateral and that the approach can be used independently or in addition to other methodologies. The recommendation is to use whatever approaches yield the best results for a particular IP.