IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 March 2024, accepted 6 May 2024, date of publication 13 May 2024, date of current version 23 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3400685

TOPICAL REVIEW

Trustworthy Integrated Circuits: From Safety to Security and Beyond

ENKELE RAMA[1,*], MOUADH AYACHE[2,3,*], RAINER BUCHTY[3], BERNHARD BAUER[4], MATTHIAS KORB[1], (Senior Member, IEEE), MLADEN BEREKOVIC[3], (Member, IEEE), AND SALEH MULHEM[3]

[1]Institute for Integrated Systems, University of the Bundeswehr Munich, 85577 Neubiberg, Germany
[2]Synopsys GmbH, 85609 Aschheim, Germany
[3]Institute of Computer Engineering, University of Lübeck, 23562 Lübeck, Germany
[4]CARIAD SE, 80807 Munich, Germany

Corresponding authors: Mouadh Ayache (mouadh.ayache@synopsys.com) and Enkele Rama (enkele.rama@unibw.de)

This work was supported in part by German Ministry of Education and Research through the publicly funded VE-VIDES Project under Grant 16ME0251.

*Enkele Rama and Mouadh Ayache are co-first authors.

ABSTRACT The trustworthiness of integrated circuits (ICs) has become increasingly important due to the ubiquitousness of ICs and the insecure nature of the current semiconductor supply chain. Throughout development and operation, ICs are exposed to several risks that can arise from malicious actors or harsh operational conditions. Therefore, the question arises: Does the trustworthiness of an IC indicate its security only or other attributes beyond? Various disciplines may have a different understanding of what *IC trustworthiness* means. Thus, a compact and unified definition that provides its main overarching attributes is required. Such a definition would lead to a greater readiness to deal with emerging challenges. To define trustworthiness at IC level, we identify the minimum number of attributes required to cover the various perspectives of development, focusing on correct functionality, reliability, security, and functional safety. Subsequently, we review and provide a structured description of identified critical pre-silicion issues that can negatively impact the defined attributes. Besides academic literature, standards, and industry-relevant publications, we consider industry experts' opinions to achieve the maximum possible coverage of our topical review. We also provide an overview and analysis of several existing evaluation methodologies of the respective trustworthiness attributes, as evaluating the discussed issues is another important aspect for achieving trustworthiness. Our findings highlight the need for a comprehensive and universally applicable framework to evaluate the trustworthiness of ICs.

INDEX TERMS EDA, integrated circuit design, reliability, safety, security, trustworthiness.

I. INTRODUCTION

Integrated Circuits (ICs) have become critical in enabling applications that affect every aspect of modern life. They are more complex and subject to stringent requirements for Power, Performance, and Area (PPA). Furthermore, new process nodes introduce additional reliability concerns. The use of ICs in security and safety-critical applications, such as data centers, automotive, healthcare, adds another layer of complexity. Meeting the requirements that arise from these challenges under tight development schedules has become increasingly difficult for IC developers. To address these various challenges, a comprehensive approach is necessary that balances these aspects throughout the IC lifecycle. Efforts to reconcile all these aspects have led to the emergence of the concept of trustworthiness.

Trustworthiness has become the backbone of several research and development strategies. For example, multiple research projects targeting trustworthy electronics have been funded in Europe, including ZEUS by the German Federal Ministry of Education and Research (BMBF) [1]. In the US, the Defense Advanced Research Projects Agency (DARPA) is funding many research projects on the system trustworthiness. In 2017, DARPA announced a five-year

The associate editor coordinating the review of this manuscript and approving it for publication was Leandros Maglaras.

plan to invest 1.5 billion US dollars in the advancement of the US semiconductor industry [2]. Furthermore, in 2022, DARPA took a step towards funding projects aimed at trustworthy artificial intelligence [3] and the trust of computing systems [4]. Therefore, a clear interpretation and formalization of trustworthiness in the context of electronics, and specifically ICs, is also required.

In their infancy, during the 1940s, digital computers were developed with largely unreliable components [5]. These components introduced faults that resulted in failures. One method of overcoming these failures was the use of redundant components. Various redundancy theories were unified under the concept of failure tolerance in 1965 [6]. Later, in 1982, a special session on fundamental concepts of fault tolerance was held at International Symposium on Fault-Tolerant Computing (FTCS)-12 [7], where various proposals were made to offer a consistent concept and terminology for fault tolerance. An encompassing concept for these proposals became necessary. Thus, between 1985 and 1992, Jean-Claude Laprie worked on developing the concept of **dependability** and led a great effort to define the basic concepts and terminology in this domain [8], [9]. In one of the first works in 1985, Laprie defined dependability to essentially encompass availability and reliability [10]. However, this definition cannot distinguish between availability and reliability in special cases, such as in the case of non-repairable systems [11], where availability reduces to reliability, as repairs after failure are not possible. Subsequently, Laprie expanded his definition of dependability by adding safety and security [9]. In [9], intentional faults, e.g., the insertion of malicious logic, were first introduced. Since then, intentional faults, as security threats, have been considered in conjunction with reliability issues. This motivated a new paradigm of computing, called dependable computing, which has been extended over the years to various domains. The Institute of Electrical and Electronics Engineers (IEEE) P2851 Working Group, which created the IEEE Standard for Functional Safety Data Format for Interoperability within the Dependability Lifecycle [12], interprets the dependability of autonomous machines as "*the property of an autonomous machine to perform reliably, safely, securely, in a time-deterministic manner, etc.*" [13]. Dependability, motivated by reliability, is an attempt to unify the various different terms, especially related to fault tolerance, into a common concept.

Trustworthiness, meanwhile, has various definitions. In some works, such as in [5], trustworthiness is considered to be equivalent to dependability, where both concepts are considered to have the same goal: "*[the] assurance that a system will perform as expected*". In other works, such as in [14], the International Federation for Information Processing (IFIP) Working Group 10.4 on Dependable Computing and Fault Tolerance uses trustworthiness to define dependability as: "*the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers*".

Still, the aspect of security is consistently incorporated into the definition of trustworthiness. Both the US National Institute for Standards and Technology (NIST) [15] and the German Federal Ministry for Economic Affairs and Energy (BMWi) [16] consider security to be the underlying prerequisite for trustworthiness. In [5], similarities and differences between dependability and trustworthiness were investigated, where threats are development, physical, and instruction faults from the perspective of dependability, while threats are hostile attacks, environmental disruptions, and human errors from the perspective of trustworthiness [17]. Although the definition of trustworthiness differs depending on the application and the defining entity, both terms are often interchangeably used, and there is no universally agreed-upon definition across domains [18].

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AoU	Assumptions of Use
ASIL	Automotive Safety Integrity Level
BIST	Build-in Self Test
CIA	Confidentiality, Integrity and Availability
CWE	Common Weakness Enumeration
DAL	Design Assurance Level
DCLS	Dual-Core Lockstep
DFA	Dependent Failure Analysis
DFT	Design for Testability
DoS	Denial of Service
DRC	Design Rule Check
EDA	Electronic Design Automation
ECU	Electronic Control Unit
FFI	Freedom from Interference
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
FMEDA	Failure Modes, Effects, and Diagnostic Analysis
FPGA	Field Programmable Gate Array
FTA	Fault Tree Analysis
HARA	Hazard Analysis and Risk Assessment
IC	Integrated Circuit
I/O	Input/Output
IoT	Internet of Things
IP	Intellectual Property
LVS	Layout Versus Schematic
OEM	Original Equipment Manufacturer
PDK	Process Design Kit
PPA	Power, Performance, and Area
RTL	Register-Transfer Level
SEooC	Safety Element out of Context
SIL	Safety Integrity Level
SM	Safety Mechanism
SoC	System-on-Chip

There are attempts to provide a comprehensive definition of trustworthiness at the *system level*. Where by system we mean "*an entity that interacts with other entities, i.e., other systems, including hardware, software, humans, and the*

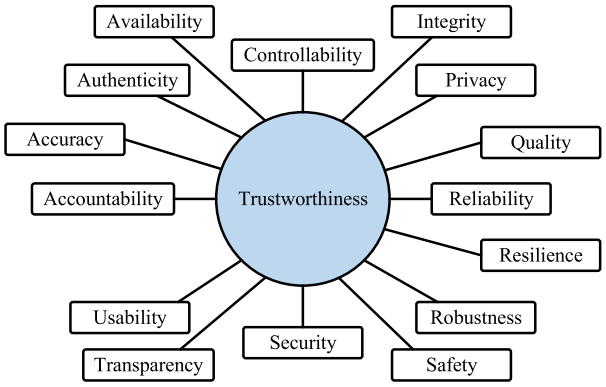

FIGURE 1. Characteristics of trustworthiness of a system per ISO/IEC TS 5723:2022.

physical world with its natural phenomena [5], *to achieve one or more stated purposes*" [19]. In electronics, such a system could be an Electronic Control Unit (ECU) used in a car. At the system level, the recent standard International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) TS 5723:2022 [19] provides a definition of trustworthiness for systems, where trustworthiness is defined as "*the ability to meet stakeholders" expectations in a verifiable way*". The various characteristics that trustworthiness can have, according to this definition, are shown in Fig. 1. Table 1, part (a), summarizes the definitions of the characteristics extracted from the ISO/IEC TS 5723:2022 [19]. In addition to the characteristics defined in the aforementioned standard, the characteristics of confidentiality, maintainability, serviceability, and repairability are included in part (b) of Table 1, since they are important in the context of electronic systems and, by extension, for ICs.

In this paper, we are specifically interested in what trustworthiness means in the context of ICs. As an IC represents a subsystem or component, we analyze trustworthiness at the *subsystem level* and investigate what can undermine it. In this case, the definition of trustworthiness relies on: (1) preventing security-related threats, such as hostile attacks [17], fault injection attacks, hardware Trojans, and Intellectual Property (IP) theft, including piracy and counterfeiting [24]; and (2) mitigating the effects of failure modes, such as complete and partial failures, or catastrophic and degraded failures [11]. Therefore, designing trustworthy ICs requires a deep understanding of security threats on the one hand and what makes a system fault-free and functionally correct on the other. In the following, we look at how trustworthiness is viewed from different perspectives.

A. TRUSTWORTHINESS OF INTEGRATED CIRCUITS FROM DIFFERENT PERSPECTIVES

Specialists who typically work independently on the main aspects of IC development: functional development, reliability, security, and functional safety, will have a different understanding of IC trustworthiness.

TABLE 1. Definition of characteristics of trustworthiness of a system.

Trustworthiness characteristic	Description
Accountability	*"State of being answerable for actions, decisions, and performance"*
Accuracy	*"Measure of closeness of results of observations, computations, or estimates to the true values or the values accepted as being true"*
Authenticity	*"Property that an entity is what it claims to be"*
Availability	*"Property of being accessible and usable on demand by an authorized entity"*
Controllability	*"Property of a system that allows a human or another external agent to intervene in the system's functioning"*
Integrity	*System integrity: "Property of accuracy and completeness"* *Data integrity: "Property whereby data have not been altered in an unauthorized manner since they were created, transmitted, or stored"*
Privacy	*"Freedom from intrusion into the private life or affairs of an individual"*
Quality	*"Degree to which a set of inherent characteristics of an object fulfills requirements"*
Reliability	*"Ability of an item to perform as required, without failure, for a given time interval, under given conditions"*
Resilience	*"Capability of a system to maintain its functions and structure in the face of internal and external change, and to degrade gracefully when this is necessary"*
Robustness	*"Ability of a system to maintain its level of performance under a variety of circumstances"*
Safety	*"Property of a system such that it does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered"*
Security	*System security: "Resistance to intentional unauthorized acts designed to cause harm or damage to a system"* *Information security: "Preservation of confidentiality, integrity, and availability of information"*
Transparency	*"Property of a system or process to imply openness and accountability"*
Usability	*"Extent to which a system product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"*

(a) Trustworthiness characteristics taken from "Trustworthiness Vocabulary" in ISO/TS 5723:2022 [19]

Maintainability	*"the probability that the system can be successfully restored to operation after failure"* [20]
Serviceability	*"relates to the time it takes to restore a system to service following a system failure"* [21]
Repairability	*"reflects the extent to which the system can be repaired in the event of a failure"* [22]
Confidentiality	*"protecting information from unauthorized access"* [23]

(b) Additional characteristics relevant for electronic systems

1) FROM THE PERSPECTIVE OF AN IC DESIGNER AND VERIFICATION ENGINEER

The goal of design and verification engineers is to ensure that requirement-fulfilling specification is implemented and that the IC function is complete and accurate. Furthermore, the design must be thoroughly verified and validated pre-silicon, as well as tested during manufacturing. This ensures that the IC behaves as expected under various operating conditions. In addition, meeting PPA requirements is essential. Thus, a trustworthy IC must not only correctly execute its functionality, but also strike a balance between power consumption, performance, and area utilization.

2) FROM THE PERSPECTIVE OF AN IC RELIABILITY ENGINEER

The goal of a reliability engineer is to ensure that the IC should function as expected without any failures during its lifecycle. This applies not only in the presence of random faults due to external sources, e.g., radiation, but also under the effects of silicon wear-out, e.g., due to aging. Therefore, it is necessary for a trustworthy IC to ensure the absence

of failures by applying measures throughout its lifecycle, including design, manufacturing, and in-field.

3) FROM THE PERSPECTIVE OF AN IC SECURITY ENGINEER

The goal of a security engineer is to implement resilient measures to protect the IC against potential security threats, such as fault injection [25], insertion of hardware Trojans [26]. Therefore, security designers and architects prioritize security threats and implement adequate countermeasures. They have gone beyond the trusted anchor paradigm and have adopted the concept of zero-trust security model *never trust, always verify* [27] from network level to hardware level [28]. Therefore, a trustworthy IC requires not only a trusted anchor, but verification and validation focused on security.

4) FROM THE PERSPECTIVE OF AN IC FUNCTIONAL SAFETY ENGINEER

The goal of a functional safety engineer is to ensure that failures in the IC do not negatively impact the safe operation of the system in which it is integrated. To be considered functionally safe and to avoid potential legal exposure in the case of catastrophic events, ICs should be developed in accordance with the applicable functional safety standards of the industry in which the IC will be used. Thus, a trustworthy IC does not cause harm during operation in the event of failure, regardless of whether the failure is the result of functional errors or reliability-driven faults.

5) INDUSTRY STANDARDS AND GUIDELINES

To align efforts and establish a common level of expectation and interoperability between the different parties involved in each specific IC development aspects, many standards and guidelines have been developed covering the various aspects. Examples of common industry standards and guidelines that apply to IC development are detailed in Table 2.

TABLE 2. Common industry standards and guidelines relevant for the discussed aspects of IC development.

Development Aspect	Standard or Guideline	Reference
Functionality and Performance	SystemC	[29]
	Unified Power Format (UPF)	[30]
	Universal Verification Methodology (UVM)	[31]
	IP-XACT	[32]
	Portable Test and Stimulus Standard (PSS)	[33]
Reliability	AEC-Q100 Failure Mechanism Based Stress Test Qualification For Integrated Circuits	[34]
	AEC-Q004 Automotive Zero Defects Framework	[35]
	SAE J1879 Handbook for Robustness Validation of Semiconductor Devices in Automotive Applications	[36]
	IEC 61709:2017 Electric Components - Reliability - Reference Conditions For Failure Rates And Stress Models For Conversion	[37]
Security	ISO/SAE 21434:2021 - Road Vehicles - Cybersecurity Engineering	[38]
	ISO/IEC 15408:2022 - Information Security, Cybersecurity and Privacy Protection	[39]
	IEC 62443 Industrial Communication Networks - Network and System Security	[40]
	ISO/IEC 19790:2012 - Information Technology - Security Techniques- Security Requirements for Cryptographic Modules	[41]
	IP Security Assurance - Security Annotation for Electronic Design Integration (SA-EDI)	[42]
Functional Safety	IEC 61508:2010 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems	[43]
	ISO 26262:2018 - Road Vehicles - Functional Safety	[44]
	IEC 61511:2016 Functional Safety - Safety Instrumented Systems for the Process Industry Sector	[45]
	IEC 61513:2011 Nuclear Power Plants - Instrumentation and Control Important to Safety	[46]
	IEC 62061:2021 Safety of Machinery - Functional Safety of Safety-related Control Systems	[47]
	DO-254/ED-80 Design Assurance Guidance for Airborne Electronic Hardware	[48]

B. MOTIVATION AND PAPER CONTRIBUTION

Due to the various perspectives described above, having a unified domain called trustworthy ICs leads to greater readiness to deal with emerging challenges. Therefore, trustworthiness at ICs level requires a compact definition with main overarching attributes. This would enable developers from different fields, who may have narrow field-specific perspectives on trustworthiness, to gain a better understanding of how trustworthiness impacts IC development overall. In this paper, we identify the minimum number of attributes required for the concept of trustworthiness at IC level. In addition, we elaborate on some of the most critical issues, spanning the various attributes, which should be addressed to achieve trustworthiness. Each issue is described in a consistent manner, focusing on the impact it has on trustworthiness and on existing and emerging countermeasures. The contributions can be listed as follows.

- We provide a compact definition of trustworthy ICs focusing on four main attributes.
- We discuss the various impairments to trustworthiness attributes and their interactions.
- We investigate critical pre-silicon issues that can negatively impact the defined trustworthiness attributes and present them in a consistent way.
- We provide an overview of several existing methodologies for evaluating trustworthiness attributes.

Instead of proposing a new taxonomy that lists and classifies all potential trustworthiness issues, in this paper, we focus on selected issues that have a critical impact on the trustworthiness attributes and hold particular practical significance. The presented issues are classes of issues gathered from discussions with IC architects, designers, and verification engineers within the scope of the VE-VIDES project for trustworthy electronics [49]. The advantage of presenting the issues in such classes is that it provides an overview from a practical perspective along the pre-silicon stages of IC development.

Throughout this paper, we consider various sources, including academic literature, standards, industry experts' opinions, and industry-relevant publications, such as white papers. To the best of our knowledge, this is the first attempt to define trustworthiness in the context of ICs, by focusing on how it affects the various different development perspectives.

C. PAPER ORGANIZATION

The remainder of this paper is organized as follows. In Section II, we elaborate on the main attributes of trustworthy ICs and explain the various related impairments and their interactions. In Section III, we describe critical issues that can negatively impact the trustworthiness of ICs. We focus on elaborating what the issue is, how, where and by whom the issue is caused, how it impacts trustworthiness, and what countermeasures can be taken to counteract it. In Section IV, we provide an overview of several existing methodologies for evaluating the specific attributes of trustworthy ICs. In Section V, we conclude and highlight the need for a unified evaluation framework for trustworthiness of ICs.

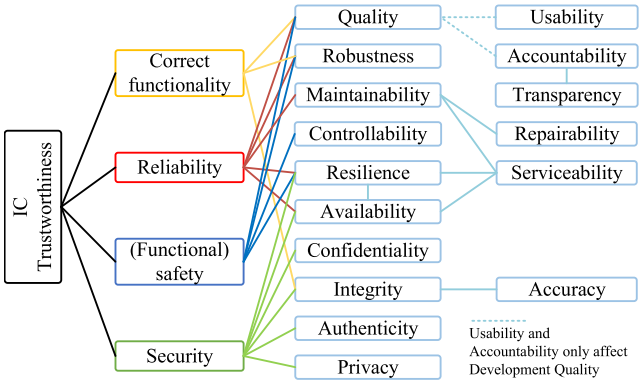

FIGURE 2. Attributes and subattributes of trustworthy ICs.

II. TRUSTWORTHY INTEGRATED CIRCUITS: ATTRIBUTES AND IMPAIRMENTS

To be able to elaborate on critical issues that negatively impact trustworthiness of ICs, it is essential to define trustworthiness in the context of ICs, as well as to identify its main attributes. Furthermore, it is important to understand the potential impairments to those attributes and the interactions between them.

A. TRUSTWORTHINESS ATTRIBUTES FOR INTEGRATED CIRCUITS

The trustworthiness of a system encompasses a set of characteristics or attributes, as described in Table 1. This can be extended or reduced depending on the type of system, its application, and the industry in which it is applied. Since an IC is considered a subsystem within an electronic system, its trustworthiness is a prerequisite for the trustworthiness of the entire system. Although, as discussed above, various attributes are related to trustworthiness, there can be a hierarchical relationship between these attributes, as many of the attributes shown in Table 1 can be assigned as subattributes of others.

Our goal is to establish the minimum number of attributes required to address the various challenges faced during IC lifecycle as a result of emerging applications. Any definition of IC trustworthiness should enable the main development teams, which work independently of each other, to have a holistic understanding of the arising challenges. Motivated by this, we consider the following four as the main IC trustworthiness attributes: (1) correct functionality, (2) reliability, (3) security, and (4) safety, as they reflect the main aspects of development. The other attributes in Table 1 can be assigned as subattributes of these four, as shown in Fig. 2.

The **correct functionality** of an IC is the ability of the IC to execute the intended functionality and only the intended functionality. Per ISO/IEC TS 5723:2022 [19], the system trustworthiness must "*meet expectations in a verifiable way*". This places the burden on the subsystems, which in the case of electronics is an ICs, to fulfill their functionality correctly. Thus, the attribute of correct IC functionality is a minimum requirement for trustworthiness. Furthermore, functionality must be verifiable, which makes it necessary to have a sufficient level of *development quality* that can assign *accountability* within the developing entity through *transparency* during development. The attribute of *usability*, at system level, is defined as the "*extent to which a system can be used to achieve specified goals with effectiveness, efficiency, and satisfaction*" [19], which is only possible if the functionality is implemented correctly at IC level. In addition, a correct function must be complete and accurate, thus covering the subattributes of *integrity* and *accuracy*. Lastly, *robustness*, which is concerned with maintaining the level of performance, is also necessary for the IC to perform the functionality correctly under varying circumstances.

The **reliability** of an IC is the ability of the IC to perform its functionality over its lifetime without failures. This covers the subattributes of *robustness*, which requires that an IC maintains its level of performance; and *resilience*, which is related to the ability of the IC to maintain its functionality regardless of internal or external changes, e.g., faults. A prerequisite of reliability in the context of ICs is *semiconductor quality*, which is concerned with early-life failures. A certain *availability* of the IC is necessary to consider it reliable. *Maintainability*, which is concerned with the probability that the system will be restored after failure, is affected by other subattributes, such as *repairability*, concerned with the extent of restoration, and *serviceability*, concerned with the restoration time. In addition, *serviceability* and *repeatability* impact several other subattributes, as shown in Fig. 2.

Safety is "*the freedom from unacceptable risk of physical injury*" [50]. In systems, such as electronic ones, when it is possible to take countermeasures to ensure that the function does not cause safety-related issues, the concept of functional safety becomes relevant. **Functional safety** is concerned with the safe functioning of a system; more specifically, "*freedom from unacceptable risk of injury or damage to people's health by properly implementing one or more automatic protection functions*" [43]. Therefore, functional safety is considered a main attribute of trustworthiness in the context of ICs, as it becomes especially relevant for safety-critical industries, such as automotive, medical, and

aerospace, where significant effort is spent, up to 60% increase [51], to ensure the functional safety of ICs. For an IC to be functionally safe, it must cover the already discussed subattributes of *development and semiconductor quality*, to address systematic and random faults; as well as *robustness*, and *resilience*. Finally, functional safety covers the subattribute of *controllability*, which is concerned with whether sufficient measures are taken so that external parties can control the impacts of a failure when necessary to maintain safety.

The **security** of an IC is the protection of the IC from unauthorized access, manipulation, or any form of malicious interference that could compromise its functionality or the confidentiality of the data it handles. A secure IC must exhibit the properties of the subattributes of *confidentiality, integrity, and availability (CIA)*, as well as provide *privacy* for the data handled by the IC. The subattribute of *confidentiality*, in the context of IC development, is not only concerned with data but also with the *confidentiality* of associated IP, e.g., design information. *Availability*, in the context of data, is also a related subattribute. Finally, a secure IC must exhibit *resilience* against attacks.

B. IMPAIRMENTS TO THE TRUSTWORTHINESS OF INTEGRATED CIRCUITS

To ensure trustworthiness, it is necessary to understand its impairments. Since IC trustworthiness encompasses multiple attributes, many impairments can negatively affect it; they can be classified as *systematic* faults, *random* faults, and *intentional* faults.

A **fault** is an abnormal condition that, when activated, can cause an **error**, which, when propagated, can lead to a failure [44]. A **failure** is defined as the loss of ability to perform a function as required [44].

A **systematic failure** is the result of deterministic **systematic faults** such as lack of systematicity during development, i.e., not following systematic approaches correctly, resulting in design mistakes. Systematic faults can also lead to security weaknesses, such as side-channel information leakage, which can be exploited later by malicious parties. This type of failure can only be prevented by applying process or design measures.

Random faults are faults that are probabilistic in nature and arise from many sources, e.g., radiation, process variation, temperature, electromagnetic interference, and transistor aging [44]. They can be classified into transient and permanent faults. Random faults can lead to **random failure** due to two types of errors: soft and hard errors. Soft errors, due to transient faults, have a transient effect on the semiconductor device that disappears by itself after some time, while hard errors, due to permanent faults, have a permanent effect that lasts indefinitely if not repaired. Soft errors are mainly the result of Single-Event Effects (SEEs) caused by particle radiation, e.g., cosmic rays and alpha particles, categorized mainly as Single-Event Upset (SEU) and Single-Event Transient (SET), depending on their effect on silicon. While SEU induces bit-flips in memory cells states directly, SET affects combinational logic causing transient voltage disturbance that can manifest as SEU in sequential elements if successfully propagated and latched, thus not electrically, logically, and temporarily masked. On the other hand, hard errors are permanent transistor damages that have different forms, e.g., stuck-at-0, stuck-at-1, or bridging faults, which are shorts between two signal lines. Permanent faults arise due to many reasons, including manufacturing defects, transistor aging effects, such as electromigration, strong radiation, and systematic faults during design [52], [53], [54].

Finally, **intentional faults** are faults introduced by a malicious entity to carry out an attack. Such faults can be active, in cases where normal operation is disrupted, or passive, in cases where inherit properties of the semiconductor are exploited.

1) HOW IMPAIRMENTS AFFECT TRUSTWORTHINESS ATTRIBUTES

In the following, we explain how the aforementioned main attributes of trustworthy ICs can be affected by the described impairments.

- The correct functionality of an IC can be negatively affected by various factors: (1) design and production mistakes (systematic faults), (2) random radiation-induced events leading to silicon faults (random faults), and (3) malicious influences, such as the insertion of Trojan circuits (intentional faults), which can violate the correct operation of an IC.
- The reliability of an IC is mainly affected by random failures that occur due to random faults, e.g., SEUs. Furthermore, reliability can be affected by early wear-out (systematic fault) in cases where reliability is not considered properly during development.
- The security of an IC is affected primarily by intentional faults. Furthermore, systematic faults, e.g., the lack of countermeasures implemented in the design, can be exploited to affect security. However, random faults are usually not relevant for security, unless they impact the functionality of parts of the IC responsible for the security functions, such as the root of trust or the Advanced Encryption Standard (AES) cipher.
- The functional safety of an IC is affected by systematic and random faults, which can lead to failures that negatively affect the safety-relevant functions, leading to violations of safety goals [44].

In summary, while systematic faults are the main impairment to correct functionality of ICs, random faults are primarily a reliability issue, which can eventually also negatively affect correct functionality. Both systematic and random faults need to be addressed to achieve functional safety, while intentional faults are the main concern for security.

C. INTERACTION BETWEEN DIFFERENT IMPAIRMENTS

Although we have already described the main attributes and impairments, there are additional terms that should be elaborated to understand trustworthiness and analyze the various issues that negatively impact it. In this section, we will elaborate on those terms and indicate how they interact with each other.

Threat describes the potential of an adversary to launch and execute an attack. A threat is closely related to two concepts. The first is **vulnerability** of the IC, which can be introduced by exploiting certain weaknesses. **Weaknesses** are hardware conditions that arise due to flaws in different development stages and that can compromise the CIA of the IC. The second is the **asset**, which represents something of value to a stakeholder, e.g., secret keys or IP details. Due to the existence of vulnerabilities, an asset is exposed to threats. Successful exploitation of a vulnerability is termed **exposure**. **Attackers** are malicious threat sources that carry out **attacks**, i.e., the action of exploiting a vulnerability to damage an asset. A vulnerable IC might be susceptible to specific **threat scenarios**. Still, these only become relevant with a worthwhile asset at the center of attention, i.e., an appropriate motivation for the attacker to exploit a vulnerability and let the threat scenario become a reality. Examples of threats include: fault injection attacks, which deliberately introduce faults in the hardware under attack, e.g., by means of voltage glitches or high-energy beams [24], to alter the behavior of the target IC; exploitation of architectural faults [24], e.g., the Rowhammer method [55]; hardware Trojans, which can be inserted through an untrustworthy manufacturing chain, potentially spanning poisoned Register-Transfer Level (RTL)-level library elements, untrustworthy design software, to untrustworthy manufacturing; IP piracy and counterfeiting; and reverse engineering.

Hazard is a potential source of harm caused by the system to the system user, as a result of an IC function deviation, **malfunctioning behavior**. This malfunctioning behavior could be the result of a random or deterministic failure, inappropriate performance, or an attack. While hazards arising from systematic failures can be counteracted by systematic development approaches; the identification and control of hazards that emerge from random hardware failures, and the mitigation of their effects, can be achieved using Safety Mechanisms (SMs).

Risk is a measure of the probability that a hazard or threat becomes an issue for the trustworthiness of an IC. In the context of security, risk is the probability of losing an asset, which is higher when there are numerous vulnerabilities and there is increased attacker capability. In the context of functional safety, risk is the probability that a hazard leads to harm. Overall, risk indicates the probability and severity level of hazard or threat. Therefore, risk is relevant to threats, hazards, and assets.

Considering trustworthiness as a whole is important because of the dependencies and interactions that the various development aspects have with each other. For example, a safety-critical IC used in an autonomous vehicle is also a security-critical one, but considering each aspect separately is not sufficient to achieve trustworthiness, as there is a link between hazards and threats. While hazard is more general and highly related to the system asset, threat reflects the risk of exploiting a vulnerability by an attacker to violate or harm the system asset. Fig. 3 illustrates the links between threat, hazard, and asset. Fundamentally, a threat may lead to a hazard in a safety-critical system, specifically when an attack affects safety goals. However, while a hazard does not pose a threat to security-critical systems, a fault can be exploited to create a vulnerability. Understanding the interaction between these impairments is crucial to achieve trustworthy ICs.

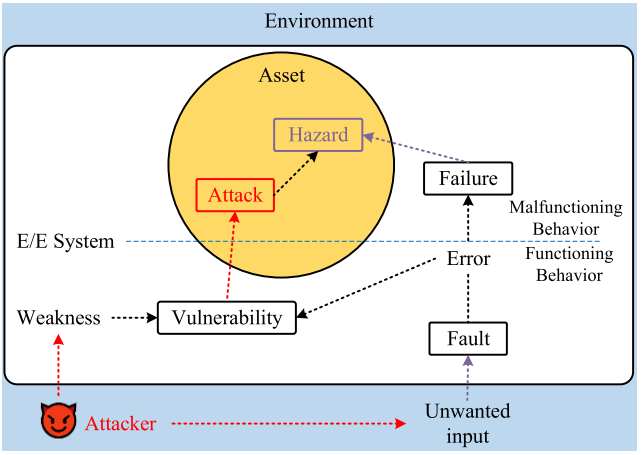

FIGURE 3. Interaction between impairments.

III. ISSUES TO THE TRUSTWORTHINESS OF INTEGRATED CIRCUITS AND THEIR IMPACT

With the fragmentation of the supply chain of ICs, ensuring the trustworthiness of each IC development stage becomes increasingly difficult, especially since various issues can negatively affect it. These issues are not only of various categories but also appear in various stages of the ICs development cycle.

Developing an IC typically starts with requirements and architecture definition, followed by front- and back-end implementation stages, and ends with the fabrication, packaging and testing stages, before being integrated into a product. A simplified model of the development cycle of an IC is shown in Fig. 4. In this paper, we focus on the issues that appear during the specification, architecture, and design stages; in other words, we focus on the pre-silicon issues to IC trustworthiness. These stages are bordered in light blue in Fig. 4.

In Section II we discussed impairments to trustworthiness attributes. There can be various causes that can appear throughout the IC development flow for these impairments. In this paper, we use the term *issue* to refer to these potential causes of IC trustworthiness impairments. The described issues are grouped in such a way as to reflect the main

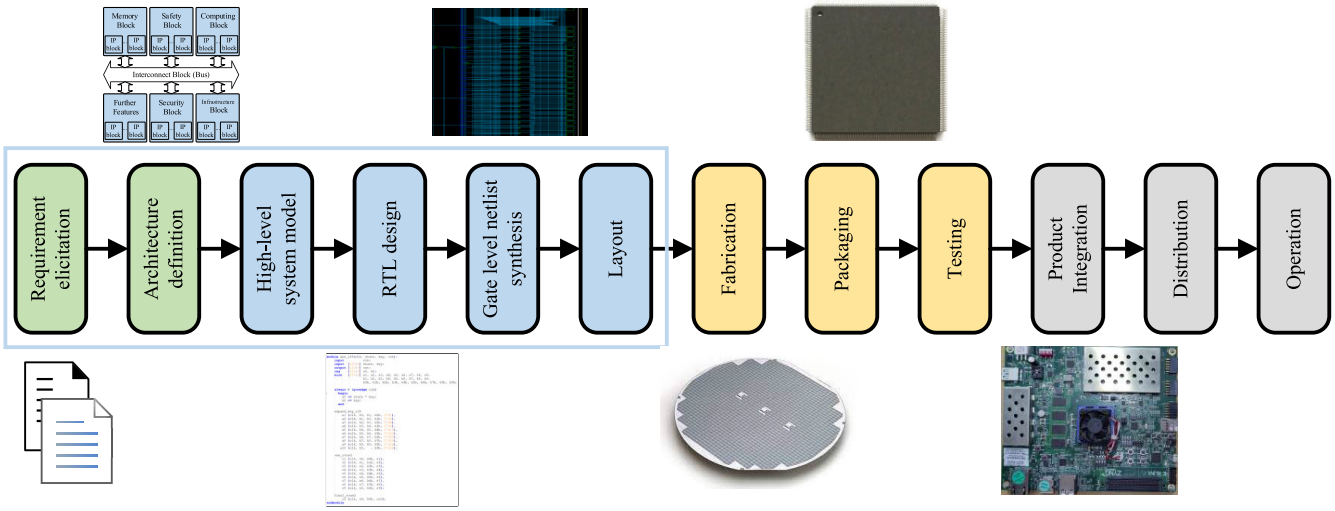

FIGURE 4. Typical IC development flow. The light blue line indicates the pre-silicon stages that are the focus of the paper.

TABLE 3. Criteria used to consistently describe issues to trustworthiness of ICs.

Criteria	Description
Internal/External	Specifies whether the issue is internal or external with respect to IC development. *Internal*: the issue originates from within development (e.g., hardware architect, IC designer, verification engineer) *External*: the issue targets the development from outside (e.g., foundry, hacker, third-party IP vendor)
Source	Identifies the source of the issue. This can be, for example, a malicious employee, a third-party IP vendor, a hacker, etc.
Stage of introduction	Specifies the stage(s) where the issue is introduced.
Outcome(s)	Specifies the potential outcome(s) from the impact of the issues.
Countermeasure(s)	Describes potential existing or emerging countermeasures to counteract the negative impacts of the issue.

attributes of trustworthiness, i.e., correct functionality, reliability, security, and functional safety. Since modern complex ICs, such as System-on-Chips (SoCs), are increasingly more dependent on externally developed third-party IP blocks, the issues that arise from their integration are treated as a separate group. The same applies to hardware Trojans, which require a dedicated focus due to their high potential for negative impact.

We describe each issue in detail, focusing on describing the issue, its impact on IC trustworthiness, and existing and emerging solutions and countermeasures. Table 3 shows the aspects that we consider when describing an issue.

To facilitate an easier reading of the paper, Fig. 5 provides an overview of the organization of Section III. Section III-A discusses issues related to the correct functionality of ICs, where Subsection III-A1 focuses on general functionality issues, while Subsection III-A2 focuses on issues associated with the integration of third-party IP blocks with relevance to functionality. Section III-B discusses reliability issues. Section III-C discusses security issues, where Subsection III-C1 focuses on general security issues, while Subsection III-C2 focuses on hardware Trojans. Finally, Section III-D discusses functional safety issues.

A. ISSUES IMPACTING CORRECT FUNCTIONALITY

In this subsection, we primarily confine ourselves to issues that negatively affect correct ICs functionality. The stages in which these issues occur are depicted in Fig. 6; where it can be seen that issues with architecture, Process Design Kit (PDK) quality, and specification of PPA parameters appear in the earlier stages of development, while those arising due to the integration of third-party IP blocks can occur at any stage, depending on the stage of the IP integration.

1) GENERAL FUNCTIONALITY ISSUES

In the following, we focus on specification-related issues appearing during the requirement elicitation and architecture definition stages that impact correct functionality of an IC. Specifically, we look at the issues caused as a result of: a flawed architecture, insufficient PDK quality, and insufficient specification of PPA parameters.

ARCHITECTURAL FLAWS

Description: The architecture of an IC is developed based on technical requirements during the architecture definition stage. The more complex ICs, such as SoCs, are generally composed of various IP blocks, also known as IP cores, or IPs for short, which are purchased from multiple IP vendors. In such an IP-based IC design methodology, the architecture definition stage becomes more important. The main advantage of following this methodology is that design houses focus on the design of innovative, stable, and efficient IC architectures and avoid the introduction of architectural flaws, which are systematic faults during architectural design, such as improper performance budgeting or the selection of unsuitable IP blocks. The main challenge with conventional

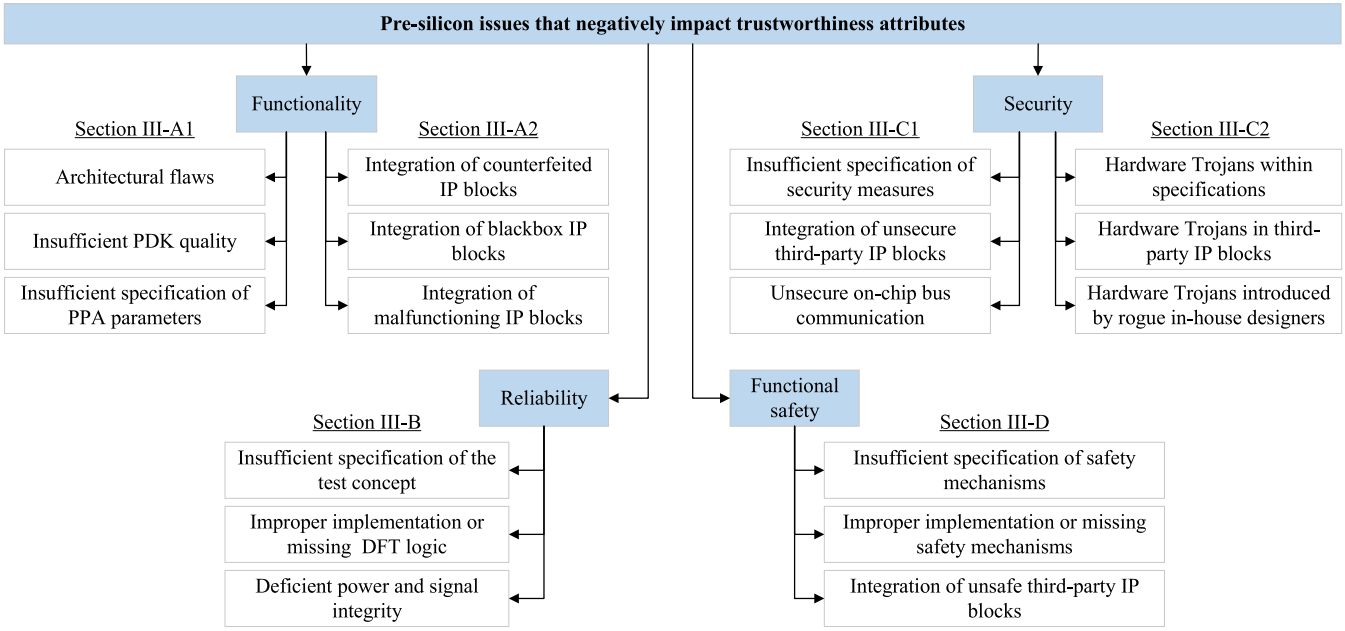

FIGURE 5. Overview of the pre-silicon issues to trustworthiness of ICs that are investigated in this paper.

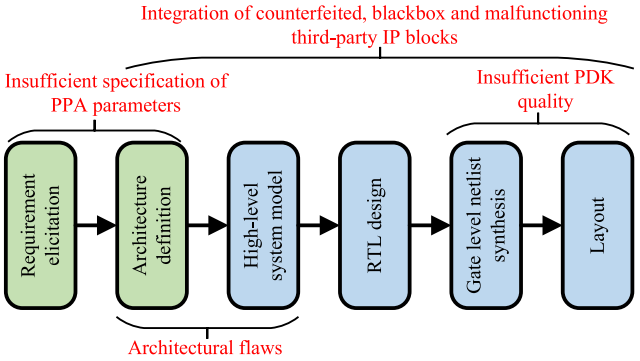

FIGURE 6. Investigated issues that affect the correct functionality of ICs.

IC architectural design is that it is a manual process that depends mainly on the skills and experience of the IC architecture team, especially when choosing appropriate IP blocks. After arriving at a suitable high-level architecture that should already fulfill the defined requirements, it is important to validate the proposed architecture against PPA requirements and other Key Performance Indicators (KPIs) by applying a virtual prototyping-based architectural analysis [56]. This process is often referred to as architectural exploration, since multiple architectures may be explored. KPI validation continues with RTL emulation, Field Programmable Gate Array (FPGA) prototyping, and ends with post-silicon testing on testers. This issue is internal to the development of the IC and can be caused by hardware architects. If not detected, the flaws may persist in the IC during deployment.

Impact on IC trustworthiness: Architectural flaws can jeopardize the functionality, performance, security, and functional safety of the IC. In fact, cost-intensive re-spins, such as the redesign of ICs after tape-out, are often the result of bugs that are found very late in the development cycle, e.g., during testing or prototyping [57].

Countermeasures: In order to avoid this issue, it is important to follow a systematic design approach and sufficient validation steps. For example, during the architecture design and verification stages, many aspects need to be carefully considered, as they have major implications on the functionality and performance of the IC. This includes hardware/software partitioning, the selection of Central Processing Unit (CPU), Graphics Processing Unit (GPU), Digital Signal Processing (DSP) cores and hardware accelerators, the size and type of memory, the type and bandwidth of on-chip interconnect, and the selection of interfaces, Input/Output (I/O) ports, and other IP blocks [56].

To arrive at a trustworthy IC architecture, it is imperative to verify the fulfillment of not only functional requirements, but also defined trustworthiness requirements. Among others, the architectures of ICs for security-critical and safety-critical applications should include dedicated components to ensure security and functional safety, such as the hardware root of trust modules [58], [59], used to provision security-critical functions, and a dedicated safety island, used to manage and monitor safety-relevant operations within the IC, as in [58] and [59]. With respect to reliability, it is vital that an appropriate process technology node that meets the failure rate requirements is selected. Similarly, appropriate IP blocks should be selected to meet reliability requirements.

Research projects invest in the establishment of standardized IC architectures to ensure trustworthiness and reduce effort at the same time. Da Silva et al. [60] have described an industry-based list of shared features for automotive SoC architectures, i.e., safety-related components,

application-specific units, automotive protocols, and security cores, and have implemented these features in an open source SoC benchmark, AutoSoC, where the notion of functional blocks is key to maintaining a modular design. Various iterations of AutoSoC have the ability to employ a variety of hardware components to meet the specific requirements of each functional block [60]. With a focus on security, there are also special standardized security architectures, e.g., ARM Trustzone [61], which establishes a Trusted Execution Environment (TEE) as an isolated part of the architecture.

INSUFFICIENT PDK QUALITY

Description: While avoiding issues with architecture definition is important, it is not sufficient if the implementation uses an insufficiently qualitative PDK. With decreasing technology node sizes and the potential usage of various technologies in a single design, such as in chiplets, PDKs get increasingly more important, while at the same time becoming more difficult to verify and synchronize, and more expensive to develop [62]. This is an external issue from the perspective of the IC developers, since PDKs are provided by the foundries.

A PDK is a set of documentation and data files that describes a fabrication process in a semiconductor foundry and enables the user to complete a design. A typical PDK contains technology files, cell libraries with models and parametric cells (pcells), rule files, verification checks, and reference flows [63], [64]. PDKs usually contains proprietary information from the foundry and trade secrets and are not always fully transparent.

Impact on IC trustworthiness: The difficulty of resolving issues that arise as a result of an insufficiently qualitative PDK increases due to the proprietary nature of PDKs. An insufficiently qualitative PDK can lead to incorrect functionality, performance and yield issues, and costly and time-consuming redesign effort.

Countermeasures: Conducting review measures during the design stages, e.g., verifying current and voltage ratings, and comparing the measured data with the provided models within the PDK, may mitigate this issue [65]. However, to decrease the burden and expenditure of the IC developer, measures undertaken during PDK development are preferred. Methodologies for PDK Quality Assurance (QA) and integrity are described in literature [66], [67] and implemented in industry [68]. For example, Global Semiconductor Alliance (GSA) has developed a checklist for PDK quality for analog/mixed signal PDKs [68]. Projects are underway, including publicly funded projects [49], to develop trustworthy PDKs. XFab, for example, has introduced an automated PDK verification flow, XVerifFlow [69], the main stages of which are shown in Fig. 7. This enables verification of various parameters of pcells, while considering parameters extracted from a specification database and Component Description Format (CDF). Furthermore, XverifFlow can be used to review netlisting procedures, device extraction, and post-layout simulation.

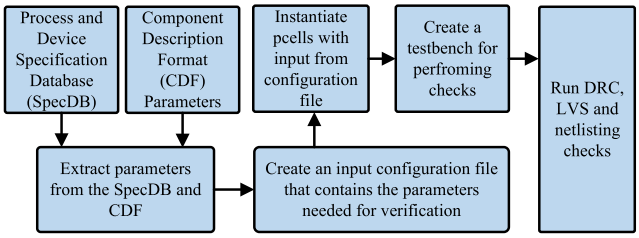

FIGURE 7. The key stages in XVerifFlow PDK verification flow adapted from [69].

INSUFFICIENT SPECIFICATION OF PPA PARAMETERS

Description: One of the main design goals is to achieve high performance at low power consumption on a small area. However, achieving this is not trivial, especially as new applications require lower area while also demanding lower power for higher performance. When considering the aim of lowering costs and increasing IC complexity, the task of PPA optimization becomes even more complex. In addition, PPA optimization can introduce bottlenecks in various aspects of the system, affecting other aspects, such as constraining the choice of technology node, IP blocks, and interconnect solutions [70]. Furthermore, the lack of accurate models, e.g., for memory utilization [70], [71], can have a significant negative impact on PPA. This is an issue that is internal from the perspective of the IC developers and can be caused by requirement engineers or hardware architects.

Impact on IC trustworthiness: Incorrect assumptions about the required power and performance can result in inadequate or faulty functionality, e.g., due to timing violations in critical paths that manifest themselves in later physical implementation stages.

Countermeasures: To address this issue, it is important to start the PPA-oriented design and verification as early as possible in the development lifecycle, also known as shift-left strategy. One such approach is to move the level of abstraction to modeling, e.g., SystemC, to allow faster development of efficient IC architectures for the target PPA [72]. This can be achieved by using virtual prototyping and architectural modeling to validate the proposed architecture against PPA requirements pre-RTL. These approaches, such as [73] and [74], can also applied to evaluate the security of ICs. Furthermore, RTL power analysis should be considered, and power validation is to be performed at later stages on the gate-level netlist and final layout. In a template-based IC design, customizing precise PPA prediction methods is essential. Tang et al. [75] propose a fast and precise PPA prediction method for template-based processor design. Furthermore, Electronic Design Automation (EDA) vendors have moved to solutions that provide faster design-space optimization and apply Artificial Intelligence (AI) [76], [77], [78]. The main idea is to learn from various design implementations and to explore various PPA combinations in parallel to find the optimum solution. For example, prior learning can vastly improve the lowest power required to

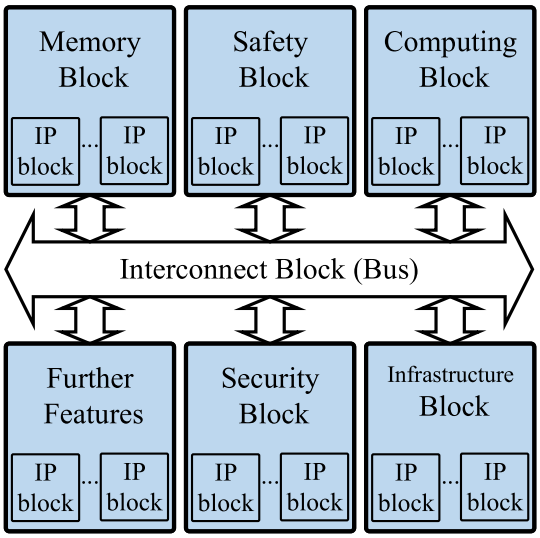

FIGURE 8. A simplified block diagram of an IC designed with the IP-based design methodology.

maintain a low Total Negative Slack (TNS) over an expert's best manual result [77].

Table 4 summarizes the general issues that affect the correct functionality of ICs.

2) ISSUES SPECIFIC TO THE INTEGRATION OF THIRD-PARTY IP BLOCKS

The integration of semiconductor IP blocks for the development of ICs has gained more adoption in recent years and has become common practice in the semiconductor industry [79], [80]. According to estimates from 2017, 75% to 80% of computer ICs included IP blocks from third-party vendors, which increased from 50% in 2013 [81]. Following this IP-based IC design methodology, a number of pre-designed blocks (i.e., processors, accelerators, memories, I/O ports and peripheral interfaces, interconnects, Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC), etc.) are integrated to reduce development cost and effort for new ICs, accelerate their development, and shorten their time-to-market [79], [82]. Such IP blocks are reusable, highly optimized, verified, and, if required for the target application, they are qualified for reliability or pre-certified for functional safety; most importantly, they are easy to integrate and configure. Due to their modularity and because they are usually deployed in previous ICs, IP integrators - a term used to describe IC design houses whose main focus is the selection, correct configuration, and integration of the purchased IP blocks to an IC, instead of the design and implementation of specific logic blocks - have more confidence in their correct functionality and assume them to be trustworthy. In addition to purchasing IP blocks from third-party IP vendors, IP blocks may be developed in-house. Fig. 8 shows a simple block diagram of an IC developed with the IP-based IC design methodology.

IP blocks can be digital, analog or mixed signal and belong to different abstraction levels, e.g., RTL (soft IPs), gate level, as a netlist (firm IPs), or layout level, in GDSII format (hard IPs) [79]. In particular, IP blocks based on standards, e.g., the Universal Serial Bus (USB) mixed-signal IP block is built following the specification of USB Implementer Forum [83], are often purchased from specialized third-party vendors. This is because such IP blocks are developed according to industry standards specifications, and developing them in-house does not provide a competitive advantage to the IP integrator. In addition, there is an increasing number of open source IPs, e.g., RISC-V cores, which are provided by the community free of licensing fees in communities, e.g., OpenCores [84] and GitHub.

In addition to cost and performance, the trustworthiness of IP blocks is a major focus of IP integrators since IP blocks are the main building block of today's complex ICs, and a single untrusted IP can compromise the trustworthiness of the entire IC, as Munsey in [85] says "a single bad IP is all it takes to break your SoC".

In the following, we focus on three different issues that affect the trustworthiness of ICs, namely the integration of counterfeited, blackbox, and malfunctioning IP blocks. From the perspective of the IP integrator, these issues represent external threats, if the integrated IP blocks are purchased from external third-party IP vendors. If the integrated IP blocks are developed in-house, then these issues may be caused by in-house IP designers, and thus may be considered internal. However, since IP blocks are usually purchased from third-party IP vendors, we consider the three aforementioned issues as external issues.

As depicted in Fig. 6, these issues can appear anywhere along the design stages of the ICs, starting with architecture definition, as they depend on the stage at which the IP is integrated, e.g. RTL for soft IPs, and layout for hard IPs. The issues described in this subsection complement the issues *Integration of unsecure third-party IP blocks*, *Hardware Trojans in third-party IP blocks*, and *Integration of unsafe third-party IP blocks*, which are mapped to the other trustworthiness attributes in Section III-C1, Section III-C2, and Section III-D, respectively.

INTEGRATION OF COUNTERFEITED IP BLOCKS

Description: The integration of counterfeited third-party IP blocks can have negative consequences for both the IP vendor and the IP integrator, depending on which party is acting maliciously. From the perspective of the IP vendor, the provided IP blocks may be extensively used by the IP integrator beyond what was agreed in the licensing agreement. This leads to the violation of the copyright of the IP vendor and to financial losses. Furthermore, the licensed IP block may be leaked or slightly modified, e.g., by adding or reducing features, without permission and then resold as a new IP block [79]. Since we focus on this paper on the impact on the trustworthiness of IC, we focus on this issue from the perspective of the IP integrator.

TABLE 4. Summary of general issues that affect the correct functionality of ICs.

Criteria	Architectural flaws	Insufficient PDK quality	Insufficient specification of PPA parameters
Internal / External	Internal	External	Internal
Source	Hardware architect	Foundry	Requirements engineer Hardware architect
Stage of introduction	Architecture definition High-level system model	Gate level netlist synthesis Layout	Requirement elicitation Architecture definition
Outcome(s)	Jeopardized functionality, functional safety, security, and performance Cost-intensive respins	Jeopardized functionality, reliability, performance and yield Cost-intensive respins	Inadequate or faulty functionality
Countermeasure(s)	Use a systematic design approach Apply sufficient validation steps Use virtual prototyping-based architectural analysis Use special standardized architectures	Apply review measures during the design stages Use PDK development checklists Utilize automated PDK verification flows	Start PPA-oriented design and verification early Use virtual prototyping and architectural modeling Use AI to optimize the design space and achieve better PPA

Counterfeited IP blocks can be acquired by IP integrators in different forms, e.g., by illegally purchasing IP blocks, which are leaked by previous buyers, disloyal employees or hackers. Furthermore, the IP integrator may integrate unauthorized IP blocks, e.g., commercial usage of academic IP blocks, or use licensed IP blocks excessively. The latter is also known as IP overuse [86], [87], [88] and can lead to legal exposure and reputational damage to the IP vendor but does not negatively impact trustworthiness assuming the IP block has not been tampered with. Counterfeited IP blocks can result from IC cloning where malicious supply chain parties clone a design in transit or reverse engineer an IC after fabrication, leading to IP theft. This issue is often referred to as IP piracy [89], [90]. The effects of the integration of counterfeited IP blocks appear during operation and are persistent, as they cannot be remedied during field operation and require the re-designing of the IC.

Impact on IC trustworthiness: Integrating counterfeited IP blocks affects all trustworthiness attributes. This is because counterfeited IP blocks may suffer from systematic faults and include malicious modifications and backdoors, e.g., due to hardware Trojans, which can lead to compromised functionality or the violation of any trustworthiness attributes. Depending on the issue that the counterfeited IP block includes, negative effects may be inflicted on any trustworthiness attribute.

Countermeasures: To reduce the risk of exposure to this issue, design houses must qualify the IP blocks they integrate for authenticity and integrity. Furthermore, IP vendors should take measures to protect their IP blocks. For example, IP watermarking techniques [91], [92], which insert a unique identify that can be checked to prove the ownership of an IP, can be applied. In addition, logic locking [93] is an active measure that locks an IP block with a secret key. This key is only known to the IP vendor and legitimate IP integrators thus preventing unauthorized usage.

INTEGRATION OF BLACKBOX IP BLOCKS

Description: It is in the interest of IP vendors to protect their copyrights to maintain a profitable business. Therefore, the IEEE has established the IEEE P1735 standard [94], which provides a unified encryption/decryption scheme and rights management for IP blocks. In this way, encrypted soft IPs are protected from direct cloning or modification of the RTL source code, while exclusively allowing functional verification and synthesis using EDA tools [79], as described in Fig. 9. However, since the RTL source code, plain text RTL, of the IP blocks is not accessible, IP integrators are forced to integrate, verify and validate them as non-transparent "blackbox" excluding the possibility of conducting RTL reviews or RTL analysis. For example, many security and trust verification techniques in the literature [95], [96], [97] require access to the RTL source code [79].

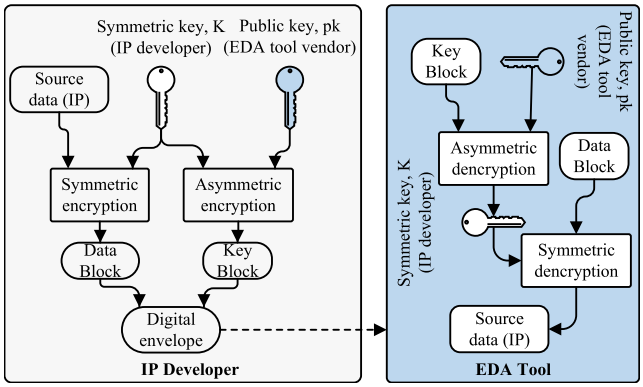

FIGURE 9. Workflow of the IEEE P1735 standard [94].

The effects of this issue are evident during verification, and the negative consequences are persistent and appear during operation.

Impact on IC trustworthiness: For IP integrators, not being able to exercise arbitrary verification and analysis techniques is a major gap that raises concerns about the trustworthiness of encrypted soft IPs. For example, such IP blocks may include security vulnerabilities, such as hardware Trojans [26], [98], [99], [100], [101], [102], [103], or functional safety issues. Depending on the problem that the blackbox IP block causes, negative outcomes may affect all trustworthiness attributes.

Countermeasures: Novel techniques that enable trust assurance despite dealing with encrypted soft IPs are needed. For example, Mishra et al. [79] suggest focusing future research on gate-level verification and analysis techniques,

since gate-level netlists, synthesized from encrypted soft IPs, are usually not encrypted. On the other hand, IP vendors offer firm or hard IPs that are usually configurable, but mostly have an inaccessible inner architecture. This leads to similar negative effects as discussed for encrypted soft IPs. In addition, integrating encrypted netlists, while cheaper, presents additional place and route, simulation, and debug challenges.

To address the challenges that arise from the integration of third-party IP blocks, especially encrypted ones, interface agreements can be set up to address liability. For example, in the automotive industry, to manage the liability of the IP integrator and final customer of ICs, the Original Equipment Manufacturer (OEM), the industry often applies Design Interface Agreements (DIAs), which is a document that defines the responsibilities of all parties, where the OEM is concerned with the system, while the suppliers are concerned with the performance and functional safety of the components [104]. Nevertheless, such agreements could be expanded to include additional checks and requirements to help increase the trustworthiness of the provided third-party IP blocks, which would be a new form of DIA, the trustworthiness DIA.

INTEGRATION OF MALFUNCTIONING IP BLOCKS

Description: To save cost, IP integrators may purchase low-cost IP blocks with potentially bad quality and functional deficiencies, which fail under specific conditions, or include bugs in their functions [105], [106]. Such issues are very hard to uncover, since IP integrators may not have technical insights or access to the internals of the IP block, as discussed for blackbox IP blocks [105], [106]. This issue is particularly dangerous, because it may be unintentionally inflicted by IP vendors, assuming that the IP vendor is not acting maliciously. Therefore, IP integrators must not assume that commercial IP blocks are bug-free and should perform functional verification of IP blocks and validation at IC-level, in addition to analyzing the quality of the purchased IP blocks before integration.

Impact on IC trustworthiness: Integrating malfunctioning IP blocks, i.e., IP blocks that do not execute their functionality as specified, or have performance issues, has a major negative impact on the functionality of the IC, such as causing inadequate and faulty functionality, as well as all other trustworthiness attributes, depending on what kind of bug or deficiency they contain.

Countermeasures: It is the responsibility of the IP integrator to integrate only high-quality IP blocks that pass a strict selection and qualification process [57]. For example, IP blocks that provide proof of proper functionality and meet stringent quality criteria should be integrated. In addition, it is important to ensure that the IP vendors have traceable and documented development processes and offer certification for their processes and IP blocks, e.g., in automotive, at least ISO 9001 [107] and International Automotive Task Force (IATF) 16949 [108] for quality, ISO 26262 [44] for functional safety, Automotive Electronics Council (AEC)-Q100 [34] for reliability qualification, ISO/Society of Automotive Engineers (SAE) 21434 for cybersecurity [38], and ISO 21448 [109] for safety of the intended functionality. Using only IP blocks that have been used previously in other ICs, can be a further criterion when integrating IP blocks in applications that require high levels of trustworthiness. However, this can have a negative impact by preventing innovation. Additionally, it is recommended to only integrate IP blocks from established IP vendors, or widely adopted open-source IP blocks since their correct functionality has been validated by the community. Finally, it is vital that the IP integrator avoids usage mistakes (1) by ensuring that the IP blocks are configured correctly and are integrated error-free; and (2) by performing IC-level validation as early as possible during development, because the later bugs are discovered, the higher the cost of fixing them [80].

Table 5 summarizes the issues related to the integration of third-party IP blocks that affect the correct functionality ICs.

B. ISSUES IMPACTING RELIABILITY

As discussed in Section II-A, reliability is an attribute of trustworthiness. Given the importance of testing to overcome issues with reliability, in this section, we focus on issues with negative impact on Design for Testability (DFT) and testing. In addition, we discuss the issue of deficient signal and power integrity, due to its impact on reliability.

DFT is an important design aspect in IC development because of the need to ensure the correct functionality and reliability of increasingly complex IC designs. In particular, in advanced technology nodes, manufacturing defects are more likely to occur, yield is lower, and semiconductors are more prone to reliability issues in the field, e.g., signal and power integrity or electromigration [110]. The goal of DFT is to allow testing during manufacturing and packaging to sort out defective dies and packaged ICs, respectively, and to enable testing during operation. Based on the test results, diagnostics and yield analysis can also be performed [111], [112], [113].

To enable the testing of ICs, DFT logic need to be inserted. For the testing of sequential elements, scan chains, which consist of a chain of flip-flops or latches that can serially shift in and out data are applied. Furthermore, Build-in Self Test (BIST) circuits, such as Logical Build-in Self Test (LBIST) and Memory Build-in Self Test (MBIST), are used during manufacturing tests and deployed during operation to monitor for permanent faults [110], [114].

DFT-related measures are involved in the different stages of IC design and manufacturing, thus issues can arise at all stages. Since the focus of this paper is on pre-silicon stages, we focus on two main issues related to DFT, i.e., the insufficient specification of the test concept, and improper implementation or missing DFT logic. In addition, we also look at the issue of deficient signal and power integrity. These

TABLE 5. Summary of issues related to integration of third-party IP blocks that affect the correct functionality of ICs.

Criteria	Integration of counterfeited IP blocks	Integration of blackbox IP blocks	Integration of malfunctioning IP blocks
Internal / External	External	External	External
Source	Third-party IP vendor	Third-party IP vendor	Third-party IP vendor
Stage of introduction	Any, starting with Architecture definition	Any, starting with Architecture definition	Any, starting with Architecture definition
Outcome(s)	Compromised functionality due to systematic faults induced during cloning or by malicious modifications and backdoors	Introduction of security vulnerabilities and functional safety issues	Inadequate and faulty functionality
Countermeasure(s)	Qualify the authenticity and integrity of integrated IP blocks Implement defense mechanisms, such as logic locking and IP watermarking	Establish DIA Use unified encryption/decryption schemes and rights management for IP blocks	Perform functional verification of IP blocks Perform validation at IC-level Analyze the quality of the IP blocks Obtain proof of proper functionality and passing quality criteria Ensure that the IP vendors have traceable and documented development processes Ensure that IP vendors offer certification for their processes and IP blocks

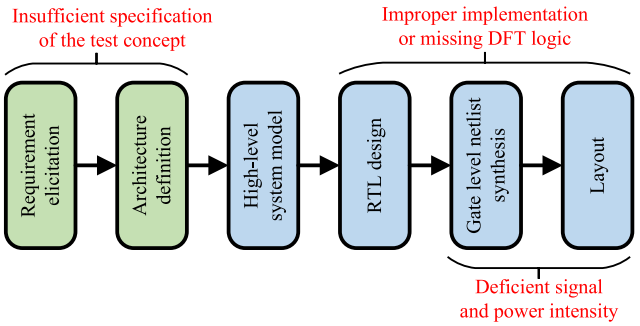

FIGURE 10. Investigated issues that affect the reliability of ICs.

issues appear at different stages of IC development, as can be seen in Fig. 10. The issues related to specification appear during the earlier specification stages, while those related to implementation begin with the RTL design stage.

INSUFFICIENT SPECIFICATION OF THE TEST CONCEPT

Description: Testability is a crucial aspect of IC development, especially due to its essential role during manufacturing and during the operation of safety-critical applications. Vague or incomplete specification of the test concept may lead to limited testability and low test coverage, and by extension to insufficient testing of ICs. The test concept is typically documented in the test plan and includes aspects of DFT methodology, the types of DFT logic in the design, the choice of test interfaces, the test strategy, and the DFT goals in terms of test coverage, yield metrics, test cost per unit, etc. This issue is persistent and is caused internally during the specification phase by requirement and DFT engineers.

Impact on IC trustworthiness: A detailed specification of the test concept is vital, not only due to the impact of testing on trustworthiness attributes such as reliability, functional safety, and security, but also to account for the high costs of testing and to avoid unnecessary silicon respins that cause enormous time and cost overhead [115].

Countermeasures: During the stages of architecture specification and definition, DFT must be considered, as it directly impacts the IC design. Modern DFT and BIST methodologies significantly increase the complexity of IC design and must be taken into account as early as possible, e.g., BIST on-chip clock control needs to be part of the IC architecture. Furthermore, compliance with industry standards, such as IEEE 1149.1 IEEE Standard for Test Access Port and Boundary-Scan Architecture, must be ensured for IC packaging. For example, four to five additional pins are needed for DFT to ensure compliance with IEEE 1149.1 [116].

IMPROPER IMPLEMENTATION OR MISSING DFT LOGIC

Description: The issue of improper implementation or missing DFT logic leads to low test coverage, which hinders yield analysis, since it cannot be ensured that yield values are accurate, and negatively affects manufacturing productivity. This issue is internal, caused, for instance, by IC designers or DFT engineers. It is persistent, since it can only be remedied by redesigning the IC, e.g., properly inserting scan chains during synthesis.

Impact on IC trustworthiness: This issue can violate all trustworthiness attributes. For example, if critical processor and memory blocks are not observed with the BIST circuits, then the functional safety of the IC cannot be ensured. Furthermore, from a reliability and functional safety perspective, DFT logic is essential to detect latent faults and defects related to aging, e.g., after power-up or through periodic self-testing [117].

For security, many DFT-related exploits that must be avoided during DFT implementation are reported in the literature, as in [118]. The full observability enabled by DFT logic is not always desired, as greater testability is known to have a negative impact on security [119], [120]. For example, scan chains can be exploited to violate confidentiality by reading out secret information, e.g., cryptographic keys, if improperly inserted in critical data paths [121]. Furthermore, the test mode can be exploited to read out the internal state of the system and reveal secrets [119], [120]. However, as discussed, low test coverage comes with the risk of delivering defective IC to customers.

Countermeasures: To ensure a sufficient implementation of DFT logic, it is vital to perform the testability analysis and coverage estimation as early as possible during the design.

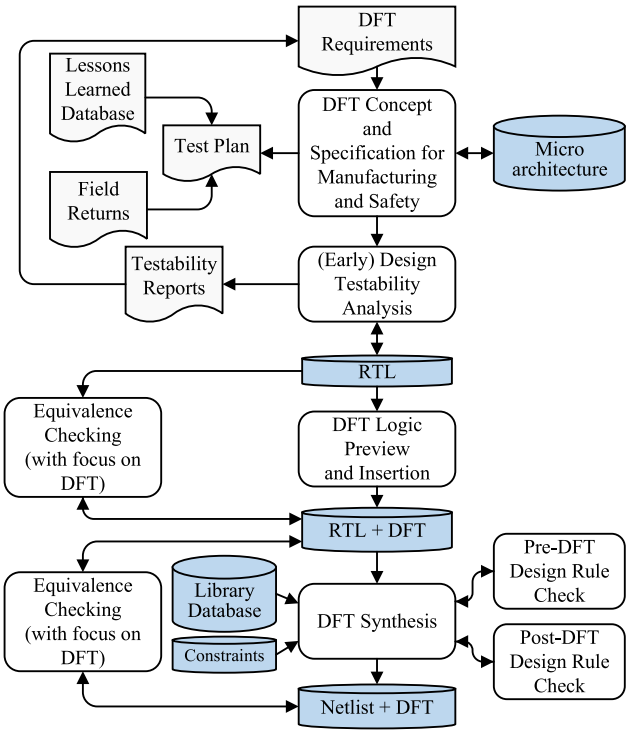

FIGURE 11. Partial flow for DFT insertion up to ATPG.

Regarding security, the integrity of critical data paths must be analyzed to ensure that sensitive information cannot be leaked through scan chains or BIST circuits. In addition, the applicable DFT and functional verification steps must be conducted after DFT insertion and synthesis, to ensure the proper implementation of DFT logic. Among others, DFT Design Rule Check (DRC) violations must be fixed, and the functional equivalence of the circuit outputs before and after DFT logic insertion must be ensured by formal verification. A suggested DFT insertion flow, up to, but not including, Automatic Test Pattern Generation (ATPG), to ensure trustworthiness for automotive applications is presented in Fig. 11. In this flow, the importance of considering DFT early in the development flow and performing design testability analysis at RTL level is reinforced.

DEFICIENT SIGNAL AND POWER INTEGRITY

Description: Signal and power integrity are significant aspects of IC design, especially for high-speed and high-performance applications. They are important to ensure that the signals and power rails in an IC meet the requirements for functionality, reliability, and performance. Specifically, signal and power integrity can become an issue when the impact of noise on timing and power is not considered in all scenarios, i.e., when the appropriate analysis does not cover all corners, functional modes, and test modes. For example, these issues can arise when the worst and most critical power cycles and events are not applied during simulation. Additional issues that affect the integrity of signals and power in an IC are the connectivity of the power mesh and the improper handling of the IR drop and electromigration. Furthermore, Electromagnetic interference (EMI), i.e., the radiation of electromagnetic fields from or into an IC due to switching currents, can cause on-chip and off-chip interference, affecting functionality and performance. EMI can also violate regulatory standards for Electromagnetic compatibility (EMC).

This issue is internal to the development of the IC and can be caused by IC designers or verification engineers who do not properly design and verify signal and power integrity.

Impact on IC trustworthiness: Having a well-designed power mesh to ensure IC signal and power integrity is essential for trustworthiness. Otherwise, an array of issues can arise, e.g., reduced performance, decreased reliability, and reduced lifetime. When applicable analysis tools for power analysis, e.g., [122] and [123], are not used during the earlier design stages, costly and time-consuming redesign efforts may be necessary.

Countermeasures: Various countermeasures can be taken to improve the integrity of signals and power in ICs. These include reducing the frequency, changing environmental parameters (voltage, temperature) of the circuit, or disabling certain functionality. These measures can be taken once the IC is deployed, but clearly, they are not preferable due to the impact on performance. Furthermore, EMI can be mitigated by using proper shielding techniques, e.g., metal enclosures or ground planes [124].

Table 6 summarizes the issues affecting IC reliability.

C. ISSUES IMPACTING SECURITY

1) GENERAL SECURITY ISSUES

As discussed, security is another important attribute of trustworthy ICs. The increase in the complexity of the design increases the attack surface for ICs and complicates the task of balancing the design for PPA and design for security [125]. Furthermore, the new technology nodes have smaller feature sizes, magnifying the effects of reliability characteristics on security. Various sources [126], [127], [128], [129], [130], [131], consider the relationship between different reliability characteristics, e.g., process variation temperature and aging, and security applications and primitives, e.g., Physically Unclonable Function (PUF) and True Random Number Generator (TRNG). Although certain characteristics, such as process variations, can be exploited for the design of PUF and TRNG, see Table 7, other reliability characteristics have a negative impact on these security primitives. Therefore, a reliability evaluation for security primitives is needed, e.g., to ensure the stability and consistency of PUF characteristics [132]. Furthermore, the increased design complexity and smaller size of the technology node features lead to an increasing number of vulnerabilities in ICs, which can result in threats to security.

Early consideration of security threats, e.g., through means of threat modeling, is essential in hardware design, especially since rectifying security gaps in the field is not straightfor-

TABLE 6. Summary of issues that affect the reliability of ICs.

Criteria	Insufficient specification of the testconcept	Improper implementation or missing DFT logic	Deficient power and signal integrity
Internal / External	Internal	Internal	Internal
Source	Requirement engineer DFT engineer Reliability engineer	IC designer DFT engineer Reliability engineer	IC designer Verification engineer Reliability engineer
Stage of introduction	Requirement elicitation Architecture definition	RTL design Gate level netlist synthesis Layout	Gate level netlist synthesis Layout
Outcome(s)	Limited testability Low test coverage Manufacturing defects Decreased reliability Silicon respins	Low test coverage Manufacturing defects Compromised functional safety Violation of confidentiality by exploiting scan chains	Reduced performance Decreased reliability Reduced lifetime
Countermeasure(s)	Prepare detailed specification Comply with relevant industry standards	Perform testability analysis and coverage estimation as early as possible Analyze critical data paths for data leakeage Apply formal verification of DFT logic insertion	Cover all corners, functional modes, and test modes during analysis Use power analysis and STA tools early in the development

TABLE 7. Impact of reliability characteristics on hardware security primitives.

Security Primitive	Reliability Characteristics				
	Process Variation	Temperature Variation	Voltage Variation	Aging	Wearout
PUF	low	high	high	high	high
TRNG	low	high	high	high	high

ward for hardware. In modern applications, e.g., in Internet of Things (IoT) or mobile phones, ICs are connected to the network, which adds an additional dimension to the attack surfaces. Hence, a preliminary step to improve security is the identification of vulnerabilities. In [133] various reporting efforts for hardware vulnerabilities are analyzed. While Common Vulnerabilities and Exposures (CVE) [134] is a database of actual publicly disclosed cybersecurity vulnerabilities, Common Weakness Enumeration (CWE) [135] is a catalog of hardware and software weakness types, and Common Attack Pattern Enumeration and Classification (CAPEC) [136] is a catalog of common attack patterns to exploit weaknesses. Although they exist, these databases are not yet fully integrated in the pre-silicon development flow for ICs. However, there are emerging solutions in the EDA industry [137] to automate the generation of security properties based on the relevant vulnerabilities contained in such databases.

The level of effort needed to ensure security is generally high, but varies depending on the specific field of application. [125]. While ICs for military applications undergo extensive security testing, those used in IoT and automotive, where PPA is critical, have lower security requirements. In the automotive IC development industry, in terms of functional safety, the relevant standard, ISO 26262-11:2018 [44], provides guidance for IC development. On the other hand, in the case of security, the relevant cybersecurity standard, ISO/SAE 21434 [38], does not provide specific guidance for the implementation of security measures during IC design or the specific verification metrics to be met, but describes the processes that must be followed at the technical and organizational level.

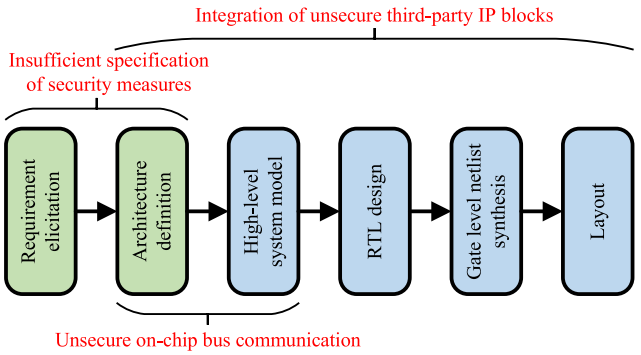

FIGURE 12. Investigated issues that affect the security of ICs.

While it is evident that there could be a multitude of threats to the trustworthiness attribute of security, in this paper, we focus on three issues, i.e., insufficient specification of security measures, integration of unsecure third-party IP blocks, and unsecure on-chip bus communication. All of these issues generally occur in the early stages of development, as depicted in Fig. 12. It should be noted that third-party IP blocks could also be integrated as hard IPs, in which case the issue would appear later during development.

INSUFFICIENT SPECIFICATION OF SECURITY MEASURES

Description: Among the main security measures applied in state-of-the-art IC design are cryptographic key generation and management, and secure communication between different IP blocks. The specification of these measures remains largely a manual effort undertaken by engineers when designing the security architecture for the IC. Insufficient specification occurs when the specification of security measures is vague or incomplete, and this can lead to security vulnerabilities. This is an internal issue, as it can be caused by the requirement engineer or hardware architect of the IC development house.

Impact on IC trustworthiness: This issue is further compounded when the insufficient specification is passed

down the implementation flow, where optimizations or incorrect implementation by the EDA tools may create new vulnerabilities [138]. The exploitation of these vulnerabilities does not only impact security itself, e.g., by leaking the cryptographic keys or sensitive data, it can also negatively impact other attributes of trustworthiness, e.g., by exploiting the vulnerabilities to cause a safety hazard in safety-critical applications.

Countermeasures: Potential countermeasures to this issue are formal approaches to executable specifications that can be easily traced throughout the development cycle. One such approach by Raj et al. [138] is Security Specification Language (SSEL), which enables the specification of protection mechanisms for various security threats. This is built on top of the programming language C, and security constructs are provided via an Application Programming Interface (API). Other approaches, such as Li et al. [139] developed Snapper, a hardware description language that "automatically inserts dynamic checks in the hardware that provably enforce a given information flow policy at execution time". Another effort by Xiao et al. seeks to reduce the manual effort needed to analyze the risks by proposing a Design Security Rule Check (DSeRC), "a framework that can be integrated into the conventional design flow to assist designers in analyzing vulnerabilities and evaluating security at all stages of the design" [140]. Given the need to balance security and PPA, any solution that does not increase design overhead and time to learn new paradigms is preferred, as it may lead to greater adoption. Furthermore, a future research direction could focus on security-driven EDA [103], where security is a constraint to consider in the development flow, in addition to functional correctness and PPA.

To avoid negative impact of security measures on functional safety, efforts are being made to ensure that the specification of security measures takes into account the functional safety implications. Among others, IEEE Standard Association has released IEEE 2851-2023 "IEEE Approved Draft Standard for Functional Safety Data Format for Interoperability within the Dependability Lifecycle", which also considers a safety-security alignment flow [12].

INTEGRATION OF UNSECURE THIRD-PARTY IP BLOCKS

Description: While in Section III-A2, we discussed the issue related to the integration of third-party IP blocks, in this section, we focus on the impact they have on IC security, for the cases where the IP blocks are not assumed to be intentionally unsecure or malevolent, e.g., contain a hardware Trojan, but can still cause security issues, e.g., leakage of cryptographic keys or sensitive data. This issue is an external threat, when looking at it from the perspective of the IP integrator, as the ICs are provided by third parties.

Impact on IC trustworthiness: The IP integrator is limited in the security evaluation that it can perform on these third-party IP blocks, because they are usually not actively involved in their development and have no access to the source of the IP block. This makes it difficult for the IP integrator to ensure that the IC is secure. The complexity increases further when the IP integrator is dealing with an encrypted IP block. Furthermore, third-party IP blocks are developed without exact knowledge of the system in which they will be integrated, which makes it is difficult for them to ensure security compliance within the IC, thus having a negative impact on IC trustworthiness.

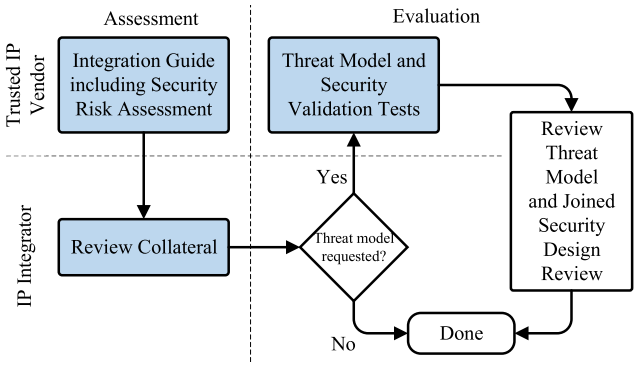

FIGURE 13. IP security assurance process flow adapted from [141].

Countermeasures: Efforts are made, such as in [141], to provide a security assurance methodology between the IP vendor and the integrator, which provides evidence for the assessment of the security risk of known relevant security concerns and an IP integration threat model, as depicted in Fig. 13.

Further efforts have also been made at the standardization level with the Security Annotation for Electronic Design Integration (SA-EDI) standard [42] by Accellera. This standard "*specifies an approach to provide information about the IP [block] security relevant to the integrator and recommended mitigations to implement and risk to address*" [42]. Based on this standard, together with the relevant design files, the IP vendor should also provide a bundle that includes: a definition of crucial assets and elements within the IP block, a database with information about security weaknesses, e.g., based on CWE, information about the behavior of the assets and the associated weaknesses, and a database with the relevant information that can be used by the IP integrator for threat modeling. This information should be exchanged in a human-readable and machine-readable format. The standard is new, first published in 2021, and efforts for its integration are underway. Since this is an external issue, a solution would be to work with proven trustworthy IP vendors, who apply the standards and flows described by the aforementioned standards.

UNSECURE ON-CHIP COMMUNICATION

Description: Another significant aspect of security, in addition to cryptographic key generation and management, is the security of on-chip bus communication between IC components. Moreover, an additional challenge appears with the increasing complexity of on-chip communication

and the utilization of Network-on-Chip (NoC) to facilitate communication between SoC modules.

Impact on IC trustworthiness: If communication between the different IC components is not protected, trustworthiness is negatively impacted, e.g., by leakage of sensitive data. For safety-critical applications, predictability, i.e., guarantees on delay and throughput, is important [142]. Attacks, such as flooding, a type of Denial of Service (DoS) attack, the NoC with additional packets [143], can cause congestion and thus affect delays, thus also affecting the functional safety of ICs deployed in a safety-critical system. This is an internal threat from the perspective of the IP integrator, and can be caused by the hardware architect.

Countermeasures: A common approach to ensure the security of IC is the integration of security subsystems [144], also known as Hardware Security Modules (HSMs). One of the main tasks of these modules is the protection of on-chip bus communication. As with other aspects of security, on-chip communication can be negatively impacted by the presence of hardware Trojans. An example is the insertion of hardware Trojans by exploiting partial specification. Fern et al. [145] developed a Trojan communication channel in an SoC bus, which is very difficult to detect and only requires altering the bus signal during a period of time that is not fully specified. Efforts to prevent this issue have also been ongoing in academia. Kim et al. [146], [147] propose a bus architecture that incorporates additional security features, e.g., security-enhanced address decoding, arbitration, multiplexing, and wrapping, to increase resilience to certain hardware Trojan attacks.

In the case of NoC, the use of more complex encryption methods in such a large design can negatively impact performance. To overcome this, Saeed et al. [148] propose a security architecture incorporating a security module for identification and verification. This approach is applicable to shared memory systems and enables secure communication by retrieving and verifying the identity of each packet. Sarihi et al. [149] propose a lightweight architecture for SoCs that utilizes an NoC that provides security against certain attacks, such as packet sniffing, with a lower area and power overhead.

Overcoming this issue is a matter of balancing the security requirements with PPA requirements. A comprehensive collection of emerging security solutions for on-chip communications is provided in [150].

Table 8 summarizes the general issues that affect the security of ICs.

2) HARDWARE TROJANS

An important category of security threats is hardware Trojans, which first appeared in the research literature about two decades ago [151]. A hardware Trojan is a modification to the circuit with malicious intentions that includes performance degradation, change in functionality, and data leakage. Generally, a hardware Trojan, a simplified representation of which is shown in Fig. 14, is made up of a trigger, used to activate the Trojan, and a payload circuit, used to implement the desired effect, where the trigger is difficult to activate and the payload is stealthy.

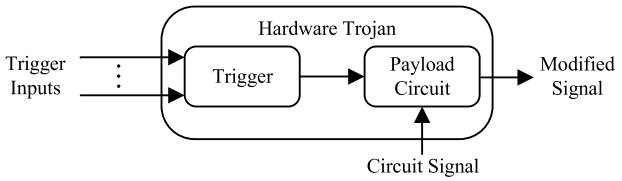

FIGURE 14. Simplified depiction of a hardware Trojan circuit.

Generally, it is assumed that hardware Trojans are inserted by third parties or untrusted actors along the development flow with the intention of harming the IP owner or the end-user [98], [99], [100], [101], [102], [103]. However, it can also be the case that the IP owner may intentionally add hardware Trojans in the form of backdoors [26], which can be used to leak data or alter functionality at a later stage during operation. Moreover, the existence of hardware Trojans affects not only security, but also all other attributes of trustworthiness. For example, hardware Trojans can be implemented to reduce reliability by accelerating wearout mechanisms [152].

Various publications exist on the taxonomy of hardware Trojans. Tehrani et al. [153] provide a summary of various Trojan taxonomies and detection methods in the initial years. Wang et al. [154] propose a taxonomy based on physical (type, size, distribution and structure), activation (externally activated and internally activated), and action (modify function, modify specification, or transmit information) characteristics. Bhunia et al. [99] propose a taxonomy based on trigger type (digital, analog) and payload type (digital, analog, information leakage and denial-of-service). Salmani et al. [155] propose a more recent and thorough taxonomy that classifies Trojans according to the insertion stage, abstraction level, activation mechanism, effect, location, and physical characteristics. Xi et al. [102] provide a similarly thorough taxonomy, synthesized from previous publications.

As is evident from the complex taxonomy of hardware Trojans, it is accepted that there is no single solution to cover all potential hardware Trojans [26], [99], and countermeasures depend on the type of Trojan being targeted. Countermeasures focus on detection, design for trust, and split manufacturing for trust [98], [99], [102]. Trojan detection techniques are generally divided into pre-silicon and post-silicon [98], [100], [101], [102], [103]. Pre-silicon techniques can be based on switching probability analysis, structural checking, and security verification [103]. Post-silicon techniques can be destructive,e.g., reverse engineering, and nondestructive, e.g., optical detection, logical testing, and side-channel signal analysis [102]. To avoid the increasing costs of dealing with Trojans in later development

TABLE 8. Summary of general issues that affect the security of ICs.

Criteria	Insufficient specification of security measures	Integration of unsecure third-party IP block	Unsecure on-chip bus communication
Internal / External	Internal	External	Internal
Source	Requirements engineer Hardware architect Security engineer	Third-party IP vendor	Hardware architect Security engineer
Stage of introduction	Requirement elicitation Architecture definition	Any, starting with Architecture definition	Architecture definition High-level system model
Outcome(s)	Leaked cryptographic keys or sensitive data Safety hazard from exploited vulnerabilities	Leaked cryptographic keys or sensitive data	Leaked cryptographic keys or sensitive data Performance impact
Countermeasure(s)	Consider safety-security alignment Apply formal approaches to executable specifications Utilize a security-driven EDA flow	Apply a security assurance methodology between the IP vendor and IP integrator Work with proven trustworthy IP vendors Apply the SA-EDI standard	Integrate HSM security subsystems which proivde for the protection of on-chip bus communication

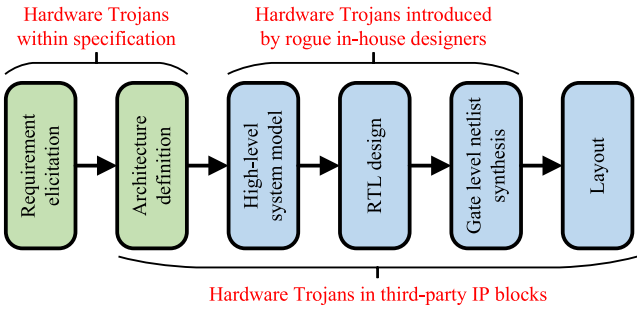

FIGURE 15. Investigated hardware Trojans in the IC development flow.

stages, when possible, pre-silicon and nondestructive techniques are preferred. Additionally, a drawback of most detection techniques is that they require a golden model, which is a Trojan-free version of the design. Since the possession of a golden model cannot always be assumed, the research trend is to move towards golden-model-free detection supported by machine learning [102]. However, machine learning algorithms themselves are also subject to attacks, where structural problems can be exploited to decrease the accuracy of classifier [156]. Trust-Hub [157] provides and maintains an increasing database of hardware Trojans that are implemented in RTL, netlist, etc.

Given the negative impact that the existence of hardware Trojans can have on the reputation of companies, information about their existence in the field is scarce; still, there are presumed examples of their impact in military applications [158]. Nevertheless, there are many efforts in academia to deal with hardware Trojans, and potential Trojans are constantly being described and analyzed.

Our proposed taxonomy for issues that affect trustworthiness of ICs covers more than just hardware Trojans. Therefore, rather than using an existing taxonomy for hardware Trojans, we focus on three practical groups of hardware Trojans introduced during the specification and design stages of IC development; i.e., hardware Trojans within specification, hardware Trojans in third-party IP blocks, and hardware Trojans introduced by rogue in-house designers. Fig. 15 shows the possible different stages in which these hardware Trojans could be inserted.

HARDWARE TROJANS WITHIN SPECIFICATION

Description: Most of the literature previously discussed focuses on hardware Trojans inserted after specification, during design and fabrication, and considers that specification is carried out by a trusted party. However, it is possible for hardware Trojans to be inserted within the specification [26], [101]. Given that most techniques for detecting hardware Trojans require a golden model that is developed based on a given specification, Trojans within specification become even more difficult to detect. Furthermore, it is possible for hardware Trojans to be inserted by exploiting a partial specification. Fern et al. [145] developed a Trojan communication channel in an SoC bus, which is very difficult to detect and only requires altering the bus signal for the period it is not fully specified. This is an internal issue from the perspective of IC development, as the hardware architect creates the specification.

Impact on IC trustworthiness: Specification Trojans can be inserted to cause inadequate library choices, data leakage, and facilitate future Trojan insertion [101]. The negative implications such Trojans have on the trustworthiness aspects of IC development are of concern. This is because the verification steps would not detect these Trojans, since they would be treated as functional features.

Countermeasures: A countermeasure to this issue is the implementation of a thorough review process of specification documents [101]. Furthermore, the use of open source IP blocks may reduce the risk of inserting hardware Trojans at this stage, due to the transparency of such IP blocks, which can make it harder for malicious actors to insert hardware Trojans without being noticed.

HARDWARE TROJANS IN THIRD-PARTY IP BLOCKS

Description: Third-party IPs represent an ideal opportunity for malicious third parties to insert design modifications and secret backdoors. From the perspective of the IP integrator, this issue represents an external threat, as the Trojans are contained in the integrated third-party IP blocks.

Impact on IC trustworthiness: The outcome of this issue can be a change in functionality, performance degradation, by causing reliability issues or even failure, data leakage,

including sensitive data, and the facilitation of future attacks [26].

Countermeasures: Various countermeasures can be applied, e.g., formal verification and code analysis [26] for soft IPs; testability-based analysis [159] for firm IPs; and post-silicon measures, such as side-channel analysis, for hard IPs [160].

HARDWARE TROJANS INTRODUCED BY ROGUE IN-HOUSE DESIGNERS

Description: An additional highly feasible strategy to introduce hardware Trojans is through rogue in-house designers [26], [161]. This issue is internal to IC development and is very difficult to defend against because it is easy for the designer to insert the Trojan during the design stages, due to their access to the internal design files.

Impact on IC trustworthiness: The outcomes of this issue are similar to those of the previously described hardware Trojans, including data leakage and IC performance degradation. What makes this type of hardware Trojans especially troublesome is the increased feasibility of attack by in-house designers since they are generally regarded as trusted parties. Furthermore, counteracting designers-inserted Trojans is even more difficult since the in-house designer is likely to have more information about the functionality of the IC, and insights about the internal development flow of the design house, and thus can tailor the Trojans to evade some verification and testing countermeasures.

Countermeasures: Among the countermeasures that could help detect this issue are code review and analysis, including formal verification, for RTL-level Trojans and Layout Versus Schematic (LVS) verification for layout-level Trojans. Furthermore, machine learning could be used for Trojan detection [162] by trying to detect abnormal design deviations.

Table 9 summarizes the issues related to hardware Trojans that affect the security of ICs.

D. ISSUES IMPACTING FUNCTIONAL SAFETY

The last aspect of trustworthiness that we consider in this paper is functional safety, which is concerned with minimizing the risk of physical injury or property damage due to the malfunction of ICs. Being functionally safe is, thus, a critical non-functional requirement for ICs to ensure their correct functionality under environmental influences, especially for safety-critical applications, such as automotive and aerospace. In fact, functional safety is the top priority for such industries, especially since they are moving towards further automation, e.g., Advanced Driver Assistance Systems (ADAS) are increasingly adapted in automotive [163] and automation is considered the future of the aviation industry [164].

Multiple industry-specific standards exist that provide guidance on how SMs, i.e., software and hardware measures to rectify the effect of random faults, should be implemented and define functional safety metrics that must be reached to achieve the required Safety Integrity Level (SIL) [50]. For the aviation industry, the DO-254/ED-80 *Design Assurance Guidance for Airborne Electronic Hardware* standards provide guidance on how to develop semiconductors and define five Design Assurance Levels (DALs) with respective functional safety metrics [48], [165]. For the automotive industry, the ISO 26262 *Road vehicles – Functional safety* standard [44] defines four Automotive Safety Integrity Levels (ASILs) and the required functional safety metrics to reach the assigned ASIL. For example, in the case of the highest level, ASIL D, these metrics need to have the following values [44]:

- Probability Metrics of Hardware Failures (PMHF) $\leq$ 10 Failure in Time (FIT);
- Single-Point Fault Metric (SPFM) $\geq$ 99%;
- Latent Fault Metric (LFM) $\geq$ 90%.

The respective ASIL is assigned based on performing Hazard Analysis and Risk Assessment (HARA) regarding the, exposure, controllability, and severity of hazards. For example, ICs used in airbags or anti-lock braking systems are very safety-critical and thus have an ASIL D rating [166].

As described in Section II-B, two main types of failures exist that can lead to safety hazards, systematic and random. SMs are essential to ensure resilience against random faults. Therefore, to achieve the required metrics and ensure the functional safety of the IC, SMs are implemented. They can be built in the form of redundant logic, such as Dual-Core Lockstep (DCLS) and Triple Modular Redundancy (TMR), or realized based on information redundancy, such as Error Correcting Code (ECC). In all cases, SMs cause an overhead in the semiconductor area and a reduction in performance, which is one of the reasons why only a necessary, but sufficient, number of SMs are implemented.

In this paper, we restrict ourselves to three main issues that are crucial to address in order to ensure the development of functionally safe ICs, i.e., insufficient specification of safety mechanisms, improper implementation or missing safety mechanisms, and integration of unsafe third-party IPs blocks. The stages in which these issues appear are depicted in Fig. 16. Although specification-related issues appear early, starting from the specification stage, those related to the implementation of safety mechanisms and the integration of third-party IPs blocks appear in later stages.

INSUFFICIENT SPECIFICATION OF SAFETY MECHANISMS

Description: The efforts to make an IC functionally safe start as early as the requirement definition and specification stage, also known as the concept phase per ISO 26262-3:2018 [44]. In the case of automotive IC development, the process of defining functional safety requirements is depicted in Fig. 17, and it involves OEM (car manufacturer) and Tier1/2 suppliers (ECU developer/IC developer). Based on HARA and the item definition, the safety goals are formulated and a functional safety concept is defined at the system level by the OEM.

TABLE 9. Summary of hardware Trojan issues that affect the security of ICs.

Criteria	Hardware Trojans within specifications	Hardware Trojans in third-party IP blocks	Hardware Trojans introduced by rogue in-house designers
Internal / External	Internal	External	Internal
Source	Requirement engineer	Third-party IP vendor	IC designer
Stage of introduction	Requirement elicitation Architecture definition	Any, starting with Architecture definition	High-level system model RTL design Gate level netlist synthesis
Outcome(s)	Inadequate library selection Sensitive data leakage Facilitation of future hardware Trojan insertion	Functionality change Performance degradation Sensitive data leakage Facilitation of future attacks	Functionality change Sensitive data leakage Performance degradation
Countermeasure(s)	Implement a thorough review process of specification documents Use of open source IP blocks	Formal verification Code analysis Testability-based analysis Side-channel analysis	Code review and analysis, including formal verification LVS verification Machine learning

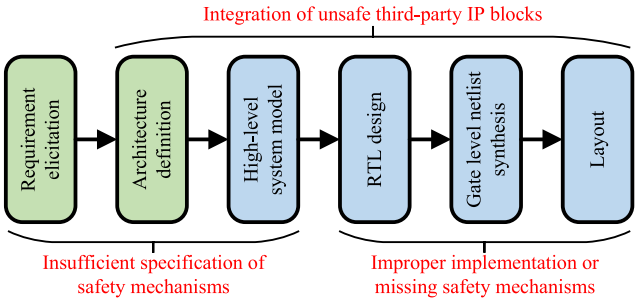

FIGURE 16. Investigated issues that affect the functional safety of ICs.

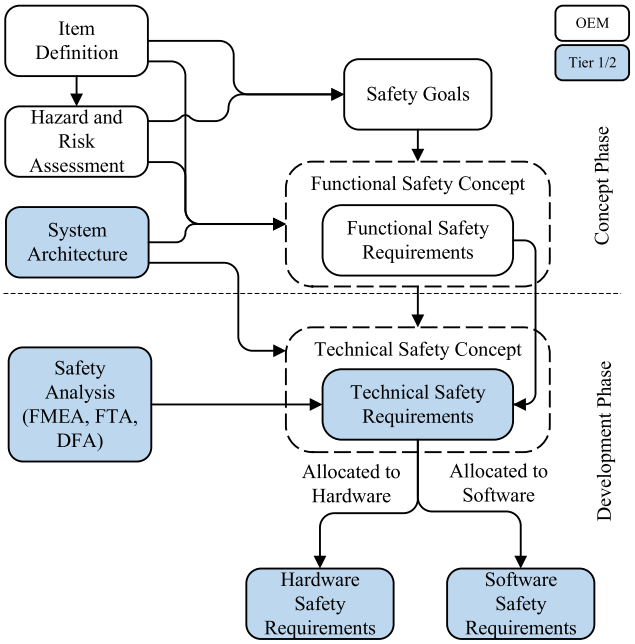

FIGURE 17. Functional safety requirements flow.

During IC development, a technical safety concept is derived by the Tier1/2, after performing a safety analysis, which considers the system architecture to arrive at the choice of the SM. The technical safety concept specifies the technical safety requirements for the architecture, which are broken down into hardware and software safety requirements that define the necessary functionalities to achieve the safety goals. Finally, the hardware development team must convert the technical safety requirements into an architecture and design that satisfy the requirements. It is possible that the ICs may also be developed by a semiconductor design house without considering a specific application, i.e., as an "off-the-shelf" generic component. In such a case, for automotive applications, ISO 26262-10:2018 provides guidance on how to develop a Safety Element out of Context (SEooC) based on Assumptions of Use (AoU) and deduced safety requirements [44], [163].

The Failure Modes and Effects Analysis (FMEA) is the most common functional safety analysis method that is performed early to identify potential failure modes and systematically analyze them with respect to their cause and consequence. The FMEA is vital for establishing safety, as it also results in the specification of the SMs. The insufficient or incorrect choice of SMs can be the result of an error in any of the earlier mentioned stages of the concept and product development phases, depicted in Fig. 17, and is internal to the IC development, under the assumption that the OEM derived functional safety requirements are complete.

Not following the latest functional safety standards is also a potential source of insufficient or wrong choice of SMs. For example, ISO 26262-5:2018 provides estimations for the diagnostic coverage of various SMs that can be considered in FMEA to choose the right combination of SMs [167].

Impact on IC trustworthiness: This issue can have safety implications, such as the selection of incorrect SMs, impacting human life and property, and can arise in many ways. Gaps and errors in the specification of SMs can arise if the FMEA is not performed completely and correctly or if the requirements specification is ambiguous. For example, critical failure modes may be left out or improper SMs may be chosen due to unclear requirements formulation.

An example of a system that failed as a result of SMs that were not properly specified is the unfortunate nuclear disaster in Fukushima, Japan. In this accident, the safety system failed even though all safety measures were activated. It was later discovered that the reason for the disaster was that the

implemented SMs did not adequately isolate the redundant system due to incorrect assumptions [168], [169], [170].

Countermeasures: To avoid this issue, many steps need to be undertaken. First, the safety specification should be complete, clear, measurable, and traceable, and should fulfill the requirements imposed by the safety standards. Furthermore, the correctness and completeness of the FMEA must be ensured.

As compliance with the standards is crucial to avoid liability, adhering to the latest versions of the standards is a must. Since semiconductor technology and the industry are constantly evolving, the safety-related values may be updated to reflect the state-of-the-art. For instance, the estimated diagnostic coverage values of SMs provided in these standards may be updated according to measured values from the field. Furthermore, an increased number of reliability issues are associated with smaller nodes leading to reduced failure rates [171]. Furthermore, ICs are being applied in new applications with potentially different environmental conditions, and the FMEA must anticipate future failure modes of these emerging applications. Finally, based on previous experiences with ICs with similar safety implications, some SMs may already be included in the functional safety concept. Such SMs must be validated during the verification stages to ensure that they apply to the current design and manufacturing technology.

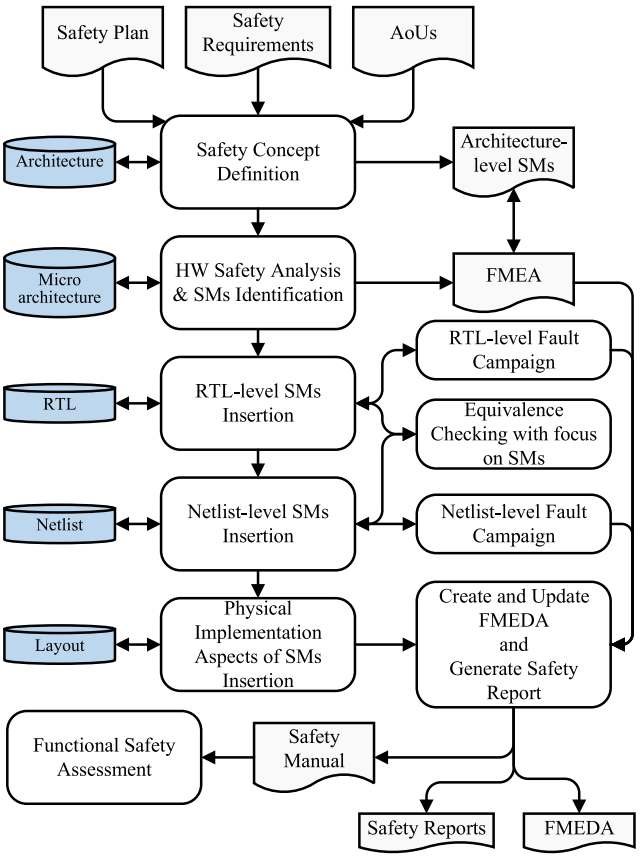

FIGURE 18. Functional safety flow, including the implementation and verification of safety mechanisms.

IMPROPER IMPLEMENTATION OR MISSING SAFETY MECHANISMS

Description: SMs can be improperly implemented by in-house designers as a result of systematic faults. Usually, SMs are implemented manually (hard coded in RTL) or semi-automatically (script-based), which requires lengthy and tedious verification and is prone to errors, especially for complex IC designs [172]. For example, a required but tedious safety verification step when implementing SMs is to ensure Freedom from Interference (FFI) of redundant blocks. This issue can arise during any stage of development and is an internal issue from the perspective of IC development.

Impact on IC trustworthiness: The outcomes of this issue are grave since the system will be unsafe and susceptible to failures that lead to hazardous events, and hence to physical injury or property damage. Furthermore, if inefficient methods are used for the implementation of SMs, there is an increase in the risk of delays in schedule, as well as of development efforts.

Countermeasures: To ensure the proper implementation of SMs, simulation-based fault injection campaigns are carried out and functional verification of SMs is performed. Using fault injection campaigns together with quantitative Failure Modes, Effects, and Diagnostic Analysis (FMEDA), it is possible to measure the diagnostic coverage of the SMs and the resulting safety metrics for the IC against random hardware failures [167]. Meanwhile, systematic failures can be avoided by using simulation-based functional verification and formal equivalence checks of pre- and post-SMs insertion netlists. Fig. 18 depicts a suggested implementation flow that considers SMs systematically from the architecture stage, e.g., using DCLS, all the way through physical implementation, e.g, ensuring FFI; as well as the respective verification stages that use fault campaigns to ensure that SMs meet the appropriate metrics at RTL and netlist-level. In addition, in the automotive domain, the ISO 26262-8:2018 standard requires that software-based tools used in the functional safety context must have a Tool Confidence Level (TCL) of 1, or otherwise be independently qualified [44].

To ensure the traceability of SMs specification throughout development, avoid manual effort, and reduce the risk of designers introducing errors during the manual implementation of SMs, the EDA industry has recently presented the possibility to automatically implement and verify SMs with EDA tools [173]. Such efforts have already begun at the standardization level by the Accellera Functional Safety Working Group [174].

INTEGRATION OF UNSAFE THIRD-PARTY IP BLOCKS

Description: Developing an IP generally involves the integration of various third-party IP blocks. It is common that not all these third-party IP blocks have been developed to

the functional safety requirements for the IC, and should be treated as SEooC. In fact, IP vendors develop their IP blocks according to the targeted ASIL and AoUs and provide the required documentation and safety manuals to the IP integrator, including the assumed safety requirements, safety concept, AoUs, FMEA/FMEDA reports, ASIL tailoring, implemented SMs, resulting functional safety metrics, etc. [44], [163], [166]. The issue of integrating unsafe third-party IP blocks arises from the risk posed by different supply chain parties, IP vendors and IP integrators, and applies to the integration of IP blocks into an IC as well as the integration of an IC into a system, e.g., an ECU. Without restricting applicability, we focus on the integration of IP blocks in the automotive domain and analyze it from the IP vendor and IP integrator perspective.

From the IP vendor perspective, targeting a high ASIL is associated with an increased overhead in the development process, especially with respect to verification and documentation. This effort should not be underestimated, especially by IP vendors who are new to the automotive domain. For example, IP vendors need to perform detailed IP-level FMEA/FMEDA that shall be provided to the IP integrator for their IC-level FMEA/FMEDA. Furthermore, depending on the type of IP block and the targeted ASIL, other functional safety analysis such as Fault Tree Analysis (FTA), Dependent Failure Analysis (DFA), and pin-level FMEA may be required [163].

This issue arises during the development stage in which the IP is integrated, and from the perspective of the IP integrator, it is an external issue, as the IP blocks are provided by third parties.

Impact on IC trustworthiness: Due to the limited availability of resources, there is a risk that IP vendors deliver IP bundles that do not include comprehensive safety reports or development documentation.

From the IP integrator perspective, blindly trusting the claims of an IP vendor without providing the required documentation and proof of meeting the respective ASIL metrics can have dangerous consequences. However, it is not feasible for the IP integrator to re-conduct the FMEA/FMEDA for all integrated IP blocks due to the lack of deep technical insights into the design of each IP block, and the limited resources. This can result in deficient IC-level FMEA/FMEDA, and hence unsafe ICs that do not mitigate the risk of all relevant failure modes to an acceptable level.

Countermeasures: To trust the safety of an IP block, the IP integrator should perform multiple steps. First of all, to meet ISO 26262 [44] requirements, IP integrators should require evidence of an organization-wide safety culture from IP vendors, and comprehensive documentation of their development process. This documentation should encompass achieved functional safety metrics, safety manuals, validation and confirmation measures, and all relevant work products related to the safety case. The safety case is defined in ISO 26262-10:2018 [44] as an "*argument that the safety requirements for an item are complete and satisfied by evidence compiled from work products of the safety activities during development*". This documentation is necessary for the safe integration of the provided IP block.

In addition to considering the provided documentation from IP vendors, it is recommended that IP integrators critically review the processes and safety culture of the IP vendor [163]. To reduce effort and accelerate the assessment, it may be reasonable to require an independent functional safety certification by dedicated parties for the processes and products of the IP integrator [44], [163], [166]. Furthermore, it remains the responsibility of the IP integrator to critically and carefully review the AoUs of each IP block to ensure that they are in line with the functional safety requirements of the IC under development.

Table 10 summarizes the issues that affect the functional safety of ICs.

IV. EVALUATION METHODOLOGIES FOR ATTRIBUTES OF TRUSTWORTHY INTEGRATED CIRCUITS

In the previous sections, various issues that negatively impact the trustworthiness of ICs were described. Although the issues described in this paper occur during the specification and design phases, the entire supply chain of ICs is susceptible to issues that affect IC trustworthiness. The described issues affect the attributes of correct functionality, security, reliability, and functional safety. In practice, these issues are handled by engineers who are focused on the discipline the issue affects. For example, a security engineer would work on implementing a secure on-chip communication system, while a functional safety engineer would work on implementing adequate SMs. Therefore, in this section, we investigate existing approaches for evaluating each of the trustworthiness attributes and explore their applicability to the issues discussed in this paper. The target being to explore whether an existing methodology covers the impact of an issue on all attributes of trustworthiness.

Currently, various trustworthiness attributes have their own attribute-specific evaluation methodologies. These methodologies can be used to assess how an issue will impact the respective development aspect of the IC. In the following, we will summarize some of the most widely used evaluation methodologies.

A. EVALUATION METHODOLOGIES FOCUSED ON RELIABILITY ASPECTS

FMEA, described in detail in Section III-D, is used in various industries, e.g., automotive, aerospace, and healthcare, for reliability purposes to identify and evaluate potential failure modes. A numerical value used for this evaluation is the Risk Priority Number (RPN). RPN is used for risk analysis and assessment, to prioritize and assess risks associated with various components. It has three factors, severity (S), occurrence (O), and detection (D), each of which can have a value of up to 10. It is calculated using (1).

$$\text{RPN} = S \cdot O \cdot D \tag{1}$$

TABLE 10. Summary of issues that affect the functional safety of ICs.

Criteria	Insufficient specification of safety mechanisms	Improper implementation or missing safety mechanisms	Integration of unsafe third-party IP blocks
Internal / External	Internal	Internal	External
Source	Requirements engineer Hardware architect Functional safety engineer	IC designer Functional safety engineer	Third-party IP vendor
Stage of introduction	Requirement elicitation Architecture definition High-level system model	Any, starting with Architecture definition	RTL design Gate level netlist synthesis Layout
Outcome(s)	Unsafe IC that can lead to the manifestation of hazardous events Impact on performance if excessive SMs are selected	Unsafe IC that can lead to the manifestation of hazardous events	IC does not mitigate the risk of all relevant failure modes to an acceptable level
Countermeasure(s)	Follow the latest relevant functional safety standard Provide all the necessary work products mandated by standards Apply functional safety-aware verification Implement sufficient SMs	Ensure FFI of redundant block Implement and verify SMs with automatically with the EDA tool flow Apply simulation-based fault injection campaigns	IP vendors should provide evidence of safety culture, documentation of their development process and the safety case IP integrator conducts its own functional safety assessment Critically and carefully review the AoUs

TABLE 11. Common Weakness Scoring System (CWSS) [176].

Base Finding	Environmental	Attack Surface
Technical Impact	Business Impact	Required Privilege
Acquired Privilege	Likelihood of Discovery	Required Privilege Layer
Acquired Privilege Layer	Likelihood of Exploit	Access Vector
Internal Control Effectiveness	External Control Effectiveness	Authentication Strenght
Finding Confidence	Prevalence	Level of Interaction
		Deployment Scope
Base Finding Subscore ⇓	**Environmental Subscore** ⇓	**Attack Surface Subscore** ⇓
CWSS Score		

By considering criticality as an additional factor, the FMEA has been extended to the Failure Modes, Effects, and Criticality Analysis (FMECA) [175], but it is not widely established for electronic systems. Since functional safety is concerned failure modes and their effect on safety-related functionalities, reliability methodologies such as the FMEA are widely adopted in functional safety standards. In addition, as described in Section III-D, FMEDA is another extension of FMEA, used in functional safety applications.

B. EVALUATION METHODOLOGIES FOCUSED ON SECURITY ASPECTS

Common Weakness Scoring System (CWSS) [176] is a security-focused scoring system that is mainly used to rate software weaknesses, but can also be applied to hardware ones. Table 11 shows the various aspects considered when calculating CWSS. It calculates three subscores for base findings, attack surface, and environmental score. For example, base finding metrics look at technical impact (potential result produced by weakness), acquired privilege (privilege obtained by attacker), acquired privilege layer (operation layer accessed by attacker), internal control effectiveness (how effectively are the internal controls against the weakness), and finding confidence (confidence that issue is a weakness and can be utilized).

The Common Weakness Risk Analysis Framework (CWRAF) [177] can be used in conjunction with CWSS to measure the risk of security weaknesses. Although it does not calculate a single metric, it does provide a structured process for evaluating risk, including a Technical Impact Scorecard, which contains the effects in case the weakness is exploited. If weaknesses lead to specific vulnerabilities, then Common Vulnerability Scoring System (CVSS) [178] can be used to assess the severity of specific vulnerabilities. This methodology can be used in security assessment, e.g., to calculate the attack feasibility when assigning a Cybersecurity Assurance Level (CAL) in the ISO/SAE 21434:2021 *Road vehicles - Cybersecurity* standard [38]. CVSS calculates a base, temporal, and environmental metric. It looks at the attack vector and complexity, the required privilege, the user interaction, and scope, while assessing the impact on CIA. The vulnerability gets a CVSS score between 0 (low) and 10 (critical), which can be calculated using the CVSS Version 3.0 Calculator [179].

Other evaluation systems also exist, including STRIDE [180] or DREAD [181]. STRIDE is focused on potential threats and attacks against software systems in order to categorize security vulnerabilities. It is used to place threats into one of the following categories: spoofing, tampering, repudiation, information disclosure, and DoS. DREAD can be used to assess the impact of such threats by assessing their damage potential, the ease of reproducibility, the level required for exportability, the number of affected users, and the ease of discoverability.

In the realm of IoT devices, especially in chip cards, an additional security evaluation methodology is provided in Common Criteria (CC) [182]. CC is a set of standards for evaluating the security features of products and systems. Among the documents published by CC, the Common Methodology for Information Technology Security Evaluation [183] defines an evaluation methodology for attack potential, which considers five different factors, as shown in Fig. 19. Each of these factors (elapsed time, specialist expertise, knowledge of item, window of opportunity, required equipment) is given a value from an adaptable numerical scale, the sum of which indicates the attack

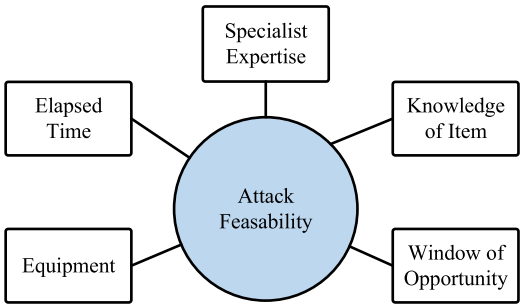

FIGURE 19. Factors considered for evaluating the attack potential per Common Criteria.

TABLE 12. Coverage of issues by existing evaluation methodologies.

Issue	Reliability related methodologies	Security related methodologies	Functional safety related methodologies
Architectural flaws		X	
Insufficient PDK quality			
Insufficient specification of PPA parameters			
Integration of counterfeited third-party IP blocks		X	X
Integration of blackbox third-party IP blocks		X	X
Integration of malfunctioning third-party IP blocks		X	X
Insufficient specification of the test concept	X		X
Improper implementation or missing DFT logic	X		X
Deficient signal and power integrity	X		X
Insufficient specification of security measures		X	
Integration of unsecure third-party IP blocks		X	
Unsecure on-chip bus communication		X	
HW Trojans within specification		X	
HW Trojans introduced in third-party IP blocks		X	
HW Trojans introduced by rogue in-house designers		X	
Insufficient specification of SMs			X
Improper implementation or missing SMs			X
Integration of unsafe third-party IP blocks			X

potential. ISO/SAE 21434:2021 Annex G.2 [38] has adapted the factors of attack potential to evaluate the feasibility of cybersecurity attacks in the automotive domain.

Since these various methodologies are focusing on the same target, i.e., reducing the risk of attacks, some factors are shared among them, such as the impact of the attack, internal controls to defend against it, and the feasibility of the attack being carried out.

C. EVALUATION METHODOLOGIES FOCUSED ON FUNCTIONAL SAFETY ASPECTS

From a functional safety perspective, there are various field-specific standards that assign a SIL. This assignment is carried out after assessing the impact of potential hazards on functional safety and conducting a risk assessment as part of the HARA process. In the automotive industry, the ISO 26262-9:2018 [44] standard provides guidance for assigning ASIL, which is described in detail in III-D. Similarly, in the aviation industry, the DO-254 *Design Assurance Guidance for Airborne Electronic Hardware* standard assigns a DAL between A (highest) and E (lowest) [184]. These methodologies consider factors such as the severity of impact, the likelihood of occurrence, and the controllability of the effects upon the occurrence of a hazard. Furthermore, as functional safety is concerned with random faults, methodologies from the reliability domain, such as FMEA, FMECA, and FMEDA may also be applied.

V. CONCLUSION

Trustworthiness is becoming an increasingly important consideration for IC development. Various disciplines may interpret this term differently; in this paper, we discuss the term trustworthiness and corresponding attributes in the context of ICs. Our aim is to provide a simplified definition based on the minimum number of attributes needed to cover all subattributes of IC trustworthiness. The four identified primary attributes are correct functionality, reliability, security, and functional safety. Moreover, we investigate the interaction between the impairments to trustworthiness attributes and perform a review of critical pre-silicon issues that can cause these impairments. Furthermore, we provide a consistent description of the issues and describe available and emerging countermeasures. Finally, we give an overview of existing methodologies dedicated to evaluating each attribute.

Although the discussed evaluation methodologies cover the effect of an issue on specific attributes of trustworthiness, such as reliability, security, and functional safety, they do not provide a unified framework for evaluating the effect of that issue on all attributes. As was evident in the description of the discussed issues, many of them have an effect on more than one attribute. However, as Table 12 shows, no existing evaluation methodology can be used to evaluate the effect of an issue on all attributes. Although certain factors are shared between the different evaluation methodologies, a nomenclature covering all these factors is lacking. Given that the breadth of issues that a trustworthy IC faces is quite broad, a comprehensive framework is needed. This topic requires further exploration in future work.

REFERENCES

[1] Bundesanzeiger. (Mar. 2020). *Vertrauenswürdige Elektronik (ZEUS)*. BMBF. Accessed: Mar. 1, 2024. [Online]. Available: https://www.elektronikforschung.de/foerderung/bekanntmachungen/zeus

[2] DARPA. (Nov. 2018). *DARPA Announces Next Phase of Electronics Resurgence Initiative*. DARPA. Accessed: Mar. 1, 2024. [Online]. Available: https://www.darpa.mil/news-events/2018-11-01a

[3] (Jun. 2022). *DARPA's ANSR to Improving Trustworthy AI*. DARPA. Accessed: Mar. 1, 2024. [Online]. Available: https://www.darpa.mil/news-events/2022-06-03

[4] (Oct. 2022). *DARPA's Selects Teams to Protect Computers' 'Roots of Trust' From Exploits*. DARPA. Accessed: Mar. 1, 2024. [Online]. Available: https://www.darpa.mil/news-events/2022-10-13

[5] A. Avižienis, J.-C. Laprie, and B. Randell, "Dependability and its threats: A taxonomy," in *Proc. Building Inf. Soc., IFIP 18th World Comput. Congr. Top. Sessions*. Toulouse, France: Springer, 2004, pp. 91–120.

[6] W. H. Pierce, *Failure-Tolerant Computer Design*. New York, NY, USA: Academic, 1965.

[7] D. Morgan, "Special session: Fundamental concepts of fault tolerance," in *Proc. 12th Annu. Int. Symp. Fault-Tolerant Comput. (FTCS)*, Jun. 1982. [Online]. Available: https://books.google.de/books/about/FTCS_12th_Annual_International_Symposium.html?id=scF3wAEACAAJ

[8] J.-C. Laprie, "Dependable computing and fault-tolerance," *Dig. Papers FTCS-15*, vol. 10, no. 2, p. 124, 1985.

[9] J.-C. Laprie, *Dependability: Basic Concepts and Terminology*. Vienna, Austria: Springer, 1992.

[10] J.-C. Laprie and A. Costes, *Dependable Computing and Fault Tolerance at LAAS: A Summary*, vol. 10. Vienna, Austria: Springer, 1985, pp. 193–213.

[11] M. Rausand and A. Hoyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, vol. 396. Hoboken, NJ, USA: Wiley, 2003.

[12] C/FSSC—Functional Safety Standards Committee, *IEEE Approved Draft Standard for Functional Safety Data Format for Interoperability Within the Dependability Lifecycle*, IEEE Standard P2851/D4.0, Jul. 2023, pp. 1–253.

[13] IEEE P2851 Working Group. *A Landscape for the Development of Dependable Machines—White Paper*. 2021, pp. 1–34. Accessed: Mar. 1, 2024. [Online]. Available: https://sagroups.ieee.org/2851/wp-content/uploads/sites/131/2021/07/Landscape_Development_Dependable_Machines.pdf

[14] IFIP Working Group 10. *Dependable computing and Fault Tolerance*. IFIP. Accessed: Mar. 1, 2024. [Online]. Available: https://www.dependability.org/?page_id=265

[15] M. Nieles, K. Dempsey, and V. Y. Pillitteri, *NIST Special Publication 800-12 Revision 1: An Introduction to Information Security*. Gaithersburg, MD, USA: NIST, Jul. 2017, doi: 10.6028/NIST.SP.800-12r1.

[16] K. Atsushi, V. Bellinghausen, and J. Fujita. (Apr. 2020). *IIoT Value Chain Security—The Role of Trustworthiness*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/IIoT_Value_Chain_Security.pdf?__blob=publicationFile

[17] F. B. Schneider, "National research council," *Trust in Cyberspace*. Washington, DC, USA: National Academy Press, 1999.

[18] B. Bauer, M. Ayache, S. Mulhem, M. Nitzan, J. Athavale, R. Buchty, and M. Berekovic, "On the dependability lifecycle of electrical/electronic product development: The dual-cone V-Model," *Computer*, vol. 55, no. 9, pp. 99–106, Sep. 2022.

[19] ISO and IEC. (2022). *ISO/IEC TS 5723:2022 Trustworthiness—Vocabulary*. International Organization for Standardization and International Electrotechnical Commission.Accessed: Mar. 1, 2024. [Online]. Available: https://www.iso.org/obp/ui/en/#iso: std:81608: en

[20] P. Sandborn and J. Myers, *Designing Engineering Systems for Sustainability*, 1st ed. London, U.K.: Springer-Verlag, 2008, p. 84.

[21] Sun Microsystems. (2006). *Netra 440 Server Product Overview—Reliability, Availability, and Serviceability Features*. Accessed: Mar. 1, 2024. [Online]. Available: https://docs.oracle.com/cd/E19102-01/n440.srvr/817-3881-12/ras.html

[22] I. Sommerville, *Chapter 10 Dependable Systems*, 10th ed. Harlow, U.K.: Pearson Education, 2019, p. 289.

[23] Office of Information Security. *Confidentiality*. Accessed: Mar. 1, 2024. [Online]. Available: https://informationsecurity.wustl.edu/items/confidentiality/

[24] J. Knechtel, E. B. Kavun, F. Regazzoni, A. Heuser, A. Chattopadhyay, D. Mukhopadhyay, S. Dey, Y. Fei, Y. Belenky, I. Levi, T. Güneysu, P. Schaumont, and I. Polian, "Towards secure composition of integrated circuits and electronic systems: On the role of EDA," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 508–513.

[25] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proc. IEEE*, vol. 100, no. 11, pp. 3056–3076, Nov. 2012.

[26] M. Xue, C. Gu, W. Liu, S. Yu, and M. O'Neill, "Ten years of hardware trojans: A survey from the attacker's perspective," *IET Comput. Digit. Techn.*, vol. 14, no. 6, pp. 231–246, Nov. 2020.

[27] S. P. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, Univ. Stirling, Stirling, Scotland, 1994.

[28] A. Deric and D. Holcomb, "Know time to die–integrity checking for zero trust chiplet-based systems using between-die delay PUFs," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. Bochum, Germany: Ruhr-Universität Bochum, 2022, pp. 391–412.

[29] IEEE, *Standard SystemC Language Reference Manual*, Institute of Electrical and Electronics Engineers, IEEE Standard 1666-2023, 2023.

[30] *Design and Verification of Low-Power, Energy-Aware Electronic Systems*, Institute of Electrical and Electronics Engineers, IEEE Standard 1801-2018, 2019.

[31] *Universal Verification Methodology Language Reference Manual*, Institute of Electrical and Electronics Engineers, IEEE Standard 1800.2-2020, 2020.

[32] *IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP Within Tool Flows*, Institute of Electrical and Electronics Engineers, IEEE Standard 1685-2022, 2023.

[33] Accellera. (2023). *Portable Test and Stimulus Standard (PSS) Version 2.1*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.accellera.org/images/downloads/standards/pss/Portable_Test_Stimulus_Standard_v2.1.pdf

[34] AEC. (Sep. 2014). *AEC—Q100: Failure Mechanism Based Stress Test Qualification for Integrated Circuits*. Automotive Electronics Council. Accessed: Mar. 1, 2024. [Online]. Available: http://www.aecouncil.com/Documents/AEC_Q100_Rev_H_Base_Document.pdf

[35] (Feb. 2020). *AEC—Q004: Automotive Zero Defects Framework*. Automotive Electronics Council, Accessed: Mar. 1, 2024. [Online]. Available: http://www.aecouncil.com/Documents/AEC_Q004_Rev-.pdf

[36] SAE International, *SAE J1879_201402: Handbook for Robustness Validation of Semiconductor Devices in Automotive Applications*, SAE International Std., 2 2014.

[37] IEC, *Electric Components—Reliability—Reference Conditions for Failure Rates and Stress Models for Conversion*, International Electrotechnical Commission, Standard IEC 61709:2017, 2017.

[38] ISO and SAE, *Road Vehicles—Cybersecurity Engineering*, International Organization for Standardization, Standard ISO/SAE 21434:2021, Aug. 2021.

[39] ISO and IEC, *Information Security, Cybersecurity and Privacy Protection*, International Organization for Standardization, Standard ISO/IEC 15408:2022, 2022.

[40] IEC, *Industrial Communication Networks—Network and System Security*, International Electrotechnical Commission, Standard IEC TS 62443, 2009.

[41] ISO and IEC, *Information Technology—Security Techniques—Security Requirements for Cryptographic Modules*, International Organization for Standardization, Standard ISO/IEC 19790:2012, 2012.

[42] Accellera. (Jul. 2021). *Security Annotation for Electronic Design Integration Standard*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.accellera.org/images/downloads/standards/Accellera_SA-EDI_Standard_v10.pdf

[43] IEC, *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, International Electrotechnical Commission, Standard IEC 61508:2010, 2010.

[44] ISO, *Road Vehicles—Functional Safety*, International Organization for Standardization, Standard ISO 26262:2018, 2018.

[45] IEC, *Functional Safety—Safety Instrumented Systems for the Process Industry Sector*, International Electrotechnical Commission, Standard IEC 61511:2016, 2016.

[46] *Nuclear Power Plants—Instrumentation and Control Important to Safety*, International Electrotechnical Commission, Standard IEC 61513:2011, 2011.

[47] *Safety of Machinery—Functional Safety of Safetyrelated Control Systems*, International Electrotechnical Commission, Standard IEC 62061:2021, 2021.

[48] RTCA, *Design Assurance Guidance for Airborne Electronic Hardware*, Radio Technical Commission for Aeronautics, Standard RTCA DO-254, Apr. 2020.

[49] Edacentrum. (2021). *Design Methods and HW/SW Co-Verification for the Unique Identifiability of Electronic Components (VE-VIDES)*. BMBF. Accessed: Mar. 1, 2024. [Online]. Available: https://www.edacentrum.de/vevides/

[50] Internation Electrotechnical Commission. *Safety and Functional Safety*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.iec.ch/functional-safety

[51] M. Tom. (Sep. 2019). *What does it cost to implement functional safety?* Accessed: Mar. 1, 2024. [Online]. Available: https://ez.analog.com/ez-blogs/b/engineerzone-spotlight/posts/what-does-it-cost-to-implement-functional-safety

[52] K. J. Hass and J. W. Ambles, "Single event transients in deep submicron CMOS," in *Proc. 42nd Midwest Symp. Circuits Syst.*, vol. 1, Aug. 1999, PP. 122–125.

[53] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.*, vol. 5, no. 3, pp. 305–316, Sep. 2005.

[54] N. Miskov-Zivanov and D. Marculescu, "Multiple transient faults in combinational and sequential circuits: A systematic approach," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 10, pp. 1614–1627, Oct. 2010.

[55] Y. Kim, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," *ACM SIGARCH Comput. Archit. News*, vol. 42, no. 3, pp. 361–372, Jun. 2014.

[56] T. Kogel, *Synopsys Virtual Prototyping for Software Development and Early Architecture Analysis*. Amsterdam, The Netherlands: Springer, 2017, pp. 1127–1159.

[57] M. Gupta. *Using 3rd Party IP in ASIC/SoC Design*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.design-reuse.com/articles/31313/using-3rd-party-ip-in-asic-soc-design.html

[58] K. Greb and D. Pradhan. *Hercules Microcontrollers: Real-Time MCUs for Safety-Critical Products*. 2011, pp. 1–11. Accessed: Mar. 1, 2024. [Online]. Available: https://www.ti.com/lit/fs/spry178/spry178.pdf

[59] S. Zhao, Q. Zhang, G. Hu, Y. Qin, and D. Feng, "Providing root of trust for ARM TrustZone using on-chip SRAM," in *Proc. 4th Int. Workshop Trustworthy Embedded Devices*, Nov. 2014, pp. 25–36.

[60] F. A. da Silva, A. Cagri Bagbaba, A. Ruospo, R. Mariani, G. Kanawati, E. Sanchez, M. S. Reorda, M. Jenihhin, S. Hamdioui, and C. Sauer, "Special session: AutoSoC—A suite of open-source automotive SoC benchmarks," in *Proc. IEEE 38th VLSI Test Symp. (VTS)*, Apr. 2020, pp. 1–9.

[61] ARM. (2009). *Building a Secure System Using TrustZone Technology*. Accessed: Mar. 1, 2024. [Online]. Available: https://documentation-service.arm.com/static/5f212796500e883ab8e74531

[62] GSA. (May 2012). *PDK Standards: A GSA White Paper and PDK Checklist*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.gsaglobal.org/wp-content/uploads/2012/05/GSAquarterlyPDKstnd.pdf

[63] L. Chrostowski and M. Hochberg, *Silicon Photonics Design: Tools and Techniques*. Cambridge, U.K.: Cambridge Univ. Press, Feb. 2015, pp. 311–312.

[64] Luceda Photonics. *PDK Structure*. Accessed: Mar. 1, 2024. [Online]. Available: https://docs.lucedaphotonics.com/reference/pdk/structure

[65] C. Urban. (Jan. 2017). *Not All Process Design Kits are Created Equal*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.intrinsix.com/blog/not-all-process-design-kits-are-created-equal

[66] M. Scott, M. Peralta, and J. Carothers, "System and framework for QA of process design kits," in *Proc. 4th Int. Symp. Qual. Electron. Design*, Mar. 2003, pp. 138–143.

[67] S. Joshi, R. Perumal, K. Gadepally, and M. Young, "An approach for a comprehensive QA methodology for the PDKs," in *Proc. 9th Int. Symp. Qual. Electron. Design (iSQED)*, Mar. 2008, pp. 480–483.

[68] *Mixed Signal PDK Quality Checklist*, GSA, Washington, DC, USA, Jul. 2013.

[69] S. Joshi and H. K. Jing. (Oct. 2021). *Introduction to an Automated PDK Cerification Flow: XVerifFlow*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.xfab.com/resourcexplorer/detail/introduction-to-an-automated-pdk-verification-flow-xverifflow

[70] C. A. Patil. (Apr. 2021). *The PPA Management in Semiconductor Product Development*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.chetanpatil.in/the-ppa-management-in-semiconductor-product-development/

[71] E. Sperling. (Oct. 2014). *3 Big Bottlenecks for Design*. Accessed: Mar. 1, 2024. [Online]. Available: https://semiengineering.com/3-big-bottlenecks-for-design/

[72] H. Schubert. (Feb. 2022). *Smaller Structures, More Complex Systems and Supply Bottlenecks*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.elektroniknet.de/international/smaller-structures-more-complex-systems-and-supply-bottlenecks.193436.html

[73] A. Nešković, S. Mulhem, A. Treff, R. Buchty, T. Eisenbarth, and M. Berekovic, "SystemC model of power side-channel attacks against AI accelerators: Superstition or not?" in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Oct. 2023, pp. 1–8.

[74] J. Treus and P. Herber, "Early analysis of security threats by modeling and simulating power attacks in SystemC," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[75] M. Tang, L. Huang, and W. Chen, "Rapid and accurate PPA prediction for the template-based processor design methods," *Appl. Sci.*, vol. 12, no. 16, p. 8383, Aug. 2022.

[76] V. Prajapati. (2019). *RTL Design Space Exploration With Oasys-RTL*. Accessed: Mar. 1, 2024. [Online]. Available: https://resources.sw.siemens.com/en-U.S./white-paper-rtl-design-space-exploration-for-best-ppa-using-oasys-rtl

[77] M. Gianfagna. (Jul. 2021). *AI and AI Chip Design—A New 'Chicken and egg' Riddle*. Accessed: Mar. 1, 2024. [Online]. Available: https://blogs.synopsys.com/from-silicon-to-software/2021/07/15/ai-chip-design-process/

[78] R. Metcalfe. (2021). *Machine Learning-Driven Full-Flow Chip Design Automation*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/digital-design-signoff/secured/cerebrus-wp.pdf

[79] P. Mishra, S. Bhunia, and M. Tehranipoor, Eds., *Hardware IP Security and Trust*. Basel, Switzerland: Springer, 2017.

[80] J. Kjelsbak. *Time to Find a Bug in a System Build Around a Big SoC*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.design-reuse.com/articles/12261/time-to-find-a-bug-in-a-system-build-around-a-big-soc.html

[81] D. J. Distel, "Markets for technology in the semiconductor industry—The role of ability-related trust in the market for IP cores," Ph.D. dissertation, Technische Universität München, Munich, Germany, 2017. Accessed: Mar. 1, 2024. [Online]. Available: https://mediatum.ub.tum.de/doc/1357158/1357158.pdf

[82] Semiconductor Engineering. *Intellectual Property (IP)*. Accessed: Mar. 1, 2024. [Online]. Available: https://semiengineering.com/knowledge_centers/intellectual-property/

[83] USB Implementers Forum. *About USB-IF*. USB Implementers Forum. Accessed: Mar. 1, 2024. [Online]. Available: https://www.usb.org/about

[84] OpenCores. *What is OpenCores?* OpenCores. Accessed: Mar. 1, 2024. [Online]. Available: https://opencores.org/

[85] M. Munsey. (Mar. 2021). *Why IP Design is Important*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.perforce.com/blog/mdx/ip-design

[86] M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits*. Basel, Switzerland: Springer, 2015, pp. 15–36.

[87] D. Forte and R. Chakraborty. (Sep. 2018). *Counterfeit Integrated Circuits: Threats, Detection, and Avoidance*. Amterdam, The Netherlands. Accessed: Mar. 1, 2024. [Online]. Available: https://ches.iacr.org/2018/slides/ches2018-tutorial1-slides.pdf

[88] S. Sikand, "Design reuse, verification reuse and dependency management," IC Manage, Inc., Campbell, CA, USA, Tech. Rep., 2013.

[89] B. Ahmed, M. K. Bepary, N. Pundir, M. Borza, O. Raikhman, A. Garg, D. Donchin, A. Cron, M. A. Abdelmoneum, F. Farahmandi, F. Rahman, and M. Tehranipoor, "Quantifiable assurance: From ips to platforms," in *Future Hardware Security Research Series*. IACR Cryptology ePrint Archive, 2022, pp. 457–533. [Online]. Available: https://eprint.iacr.org/2021/1654

[90] H. A. Shaikh, M. B. Monjil, S. Chen, F. Farahmandi, N. Asadizanjani, M. Tehranipoor, and F. Rahman, "Digital twin for secure semiconductor lifecycle management: Prospects and applications," IACR Cryptol. ePrint Archive, Tech. Rep. Paper 2022/258, 2022. Accessed: Mar. 1, 2024. [Online]. Available: https://eprint.iacr.org/2022/258

[91] R. Karmakar and S. Chattopadhyay, "Hardware IP protection using logic encryption and watermarking," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2020, pp. 1–10.

[92] A. Cui, C. H. Chang, and S. Tahar, "IP watermarking using incremental technology mapping at logic synthesis level," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 9, pp. 1565–1570, Sep. 2008.

[93] H. M. Kamali, K. Z. Azar, F. Farahmandi, and M. Tehranipoor, "Advances in logic locking: Past, present, and prospects," IACR Cryptol. ePrint Arch., Tech. Rep. Paper 2022/260, 2022. Accessed: Mar. 1, 2024. [Online]. Available: https://eprint.iacr.org/2022/260

[94] IEEE, *IEEE Approved Draft Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP)—Corrigendum 1: Correction to Rights Digest Description*, Standard P1735_Cor1/D2, Apr. 2015, pp. 1–10.

[95] M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith, "Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 159–172.

[96] X. Guo, R. Dutta, Y. Jin, F. Farahmandi, and P. Mishra, "Pre-silicon security verification and validation: A formal perspective," in *Proc. 52nd Annu. Design Automat. Conf.*, 2015, pp. 1–6.

[97] Y. Huang, S. Bhunia, and P. Mishra, "MERS: Statistical test generation for side-channel analysis based Trojan detection," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, 2016, pp. 130–141.

[98] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, "Towards Trojan-free trusted ICs: Problem analysis and detection scheme," in *Proc. Conf. Design, Automat. Test Eur.*, 2008, pp. 1362–1365.

[99] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proc. IEEE*, vol. 102, no. 8, pp. 1229–1247, Aug. 2014.

[100] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Trans. Design Autom. Electron. Syst.*, vol. 22, no. 1, pp. 1–23, Jan. 2017.

[101] I. Polian, G. T. Becker, and F. Regazzoni. (2016). *Trojans in Early Design Steps—An Emerging Threat*. Accessed: Mar. 1, 2024. [Online]. Available: https://upcommons.upc.edu/bitstream/handle/2117/99414/FCTRU_2016_55_Trojans_in_Early.pdf

[102] C. Dong, Y. Xu, X. Liu, F. Zhang, G. He, and Y. Chen, "Hardware trojans in chips: A survey for detection and prevention," *Sensors*, vol. 20, no. 18, p. 5165, Sep. 2020.

[103] W. Hu, C.-H. Chang, A. Sengupta, S. Bhunia, R. Kastner, and H. Li, "An overview of hardware security and trust: Threats, countermeasures, and design tools," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1010–1038, Jun. 2021.

[104] E. Helmig. *ISO 26262—Functional Safety in Personal Vehicles: Responsibilities and Liabilities of Functional Safety Managers*. 2021, pp. 1–15. Accessed: Mar. 1, 2024. [Online]. Available: https://www.ra-helmig.de/fileadmin/docs/publikationen/ISO_26262_Liability_Functional_Safety_Managers.pdf

[105] B. Tan, R. Elnaggar, J. M. Fung, R. Karri, and K. Chakrabarty, "Toward hardware-based IP vulnerability detection and post-deployment patching in systems-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1158–1171, Jun. 2021.

[106] S. Barrick. *Designing Around an Encrypted Netlist: Is the Pain Worth the Gain?* Accessed: Mar. 1, 2024. [Online]. Available: https://www.design-reuse.com/articles/18205/encrypted-netlist.html

[107] ISO, *Quality Management*, International Organization for Standardization, Standard ISO 9001:2015, 2015.

[108] IATF, *Quality Systems—Automotive Suppliers*, International Automotive Task Force, Standard IATF 1694:2016, Oct. 2016.

[109] ISO, *Road Vehicles—Safety of the Intended Functionality*, International Organization for Standardization, Standard ISO 21448:2022, 2022.

[110] L.-T. Wang, C. E. Stroud, and N. A. Touba, Eds., *System-on-Chip Test Architectures: Nanometer Design for Testability*. Amsterdam, The Netherlands: Elsevier, 2008.

[111] G. Eide and D. Appello. (Dec. 2009). *The Changing Role of Diagnosis in Yield Analysis*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.edn.com/the-changing-role-of-diagnosis-in-yield-analysis/

[112] Synopsys. *Yield Explorer Datasheet*. pp. 1–4. Accessed: Mar. 1, 2024. [Online]. Available: https://www.synopsys.com/content/dam/synopsys/silicon/datasheets/YieldExplorer-ds.pdf

[113] DR Yield. (Dec. 2021). *Why Should Test and Yield Engineers Use a Yield Analysis Software for Yield Diagnostics?* Accessed: Mar. 1, 2024. [Online]. Available: https://dryield.com/test-and-yield-engineers-use-yield-analysis-software

[114] C. E. Stroud, *A Designer's Guide to Built-In Self-Test*. Norwell, MA, USA: Kluwer, May 2002.

[115] S. Diamantidis, I. Diamantidis, and T. Oikonomou. *A Unified DFT Verification Methodology*. Mar. 1, 2024. [Online]. Available: https://www.design-reuse.com/articles/12633/a-unified-dft-verification-methodology.html

[116] R. Watt. Jan. 2009. *AN3812: Architecting DFT into board design to Leverage Board-Level Boundary Scan*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN3812.pdf

[117] NXP Semiconductors. (Jun. 2017). *AN11993: Using the Built-in Self Test (BIST) on the MPC5744P*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN11993.pdf

[118] G. K. Contreras, A. Nahiyan, S. Bhunia, D. Forte, and M. Tehranipoor, "Security vulnerability analysis of design-for-test exploits for asset protection in SoCs," in *Proc. 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 617–622.

[119] M. Karmani, N. Benhadjyoussef, B. Hamdi, and M. Machhout, "The DFA/DFT-based hacking techniques and countermeasures: Case study of the 32-bit AES encryption crypto-core," *IET Comput. Digit. Techn.*, vol. 15, no. 2, pp. 160–170, 2021.

[120] X. Li, W. Li, J. Ye, H. Li, and Y. Hu, "Scan chain based attacks and countermeasures: A survey," *IEEE Access*, vol. 7, pp. 85055–85065, 2019.

[121] W. Wang, X. Wang, J. Wang, N. N. Xiong, S. Cai, and P. Liu, "Ensuring cryptography chips security by preventing scan-based side-channel attacks with improved DFT architecture," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 3, pp. 2009–2023, Mar. 2022.

[122] Synopsys. *Primepower RTL to Signoff Power Analysis*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.synopsys.com/implementation-and-signoff/signoff/primepower.html

[123] *Primetime Static Timing Analysis*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.synopsys.com/implementation-and-signoff/signoff/primetime.html

[124] Thivakaran and P. J. Chong. (2022). *How to Prevent Electromagnetic Interference From Ruining Your Devices*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.ttelectronics.com/blog/electromagnetic-interference/

[125] E. Worthman. (Feb. 2015). *Challenges for IC Security*. Accessed: Mar. 1, 2024. [Online]. Available: https://semiengineering.com/challenges-for-ic-security/

[126] M. Mustapa and M. Niamat, "Temperature, voltage, and aging effects in ring oscillator physical unclonable function," in *Proc. IEEE IEEE 17th Int. Conf. High Perform. Comput. Commun. 7th Int. Symp. Cyberspace Saf. Secur., IEEE 12th Int. Conf. Embedded Softw. Syst.*, Aug. 2015, pp. 1699–1702.

[127] V. Vivekraja and L. Nazhandali, "Circuit-level techniques for reliable physically uncloneable functions," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jul. 2009, pp. 30–35.

[128] MD. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant RO-PUF for reliable key generation," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 3, pp. 335–348, Jul. 2016.

[129] R. Maes and V. van der Leest, "Countering the effects of silicon aging on SRAM PUFs," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, May 2014, pp. 148–153.

[130] N. Karimi, J.-L. Danger, and S. Guilley, "Impact of aging on the reliability of delay PUFs," *J. Electron. Test.*, vol. 34, pp. 571–586, Aug. 2018.

[131] F. Rahman, D. Forte, and M. M. Tehranipoor, "Reliability vs. security: Challenges and opportunities for developing reliable and secure integrated circuits," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2016, pp. 4C-6-1–4C-6-10.

[132] B. Halak, M. Zwolinski, and M. S. Mispan, "Overview of PUF-based hardware security solutions for the Internet of Things," in *Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Oct. 2016, pp. 1–4.

[133] J. Bellay, D. Forte, R. Martin, and C. Taylor, "Hardware vulnerability description, sharing and reporting: Challenges and opportunities," in *GOMACTech—Collaboration Amidst Isolation: Microelectron. Enabling Our Connected Nation*. Alexandria, VA, USA: National Science Foundation, Jan. 2021. Accessed: Mar. 1, 2024. [Online]. Available: https://par.nsf.gov/biblio/10237521

[134] CVE Program. *About the Common Vulnerabilities and Exposures (CVE) Program*. MITRE. Accessed: Mar. 1, 2024. [Online]. Available: https://www.cve.org/about/overview

[135] MITRE. *About Common Weakness Enumeration (CWE)*. Accessed: Mar. 1, 2024. [Online]. Available: https://cwe.mitre.org/about/index.html

[136] *About Common Attack Pattern Enumerations and Classifications (CAPEC)*. MITRE. Accessed: Mar. 1, 2024. [Online]. Available: https://capec.mitre.org/about/index.html

[137] S. Marchese. (Dec. 2020). *Make Hardware Strong With CWE*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.onespin.com/resources/blog/details/make-hardware-strong-with-cwe-1

[138] K. Raj, A. Hegde, A. P. Deb Nath, S. Bhunia, and S. Ray, "SSEL: An extensible specification language for SoC security," in *Proc. Asian Hardw. Oriented Secur. Trust Symp. (AsianHOST)*, Dec. 2021, pp. 1–6.

[139] X. Li, V. Kashyap, J. K. Oberg, M. Tiwari, V. R. Rajarathinam, R. Kastner, T. Sherwood, B. Hardekopf, and F. T. Chong, "Sapper: A language for hardware-level security policy enforcement," *ACM SIGARCH Comput. Archit. News*, vol. 42, no. 1, p. 97—112, 2 2014.

[140] K. Xiao, A. Nahiyan, and M. Tehranipoor, "Security rule checking in IC design," *Computer*, vol. 49, no. 8, pp. 54–61, Aug. 2016.

[141] B. Sherman, M. Borza, B. Rosenberg, and C. Qi, "Security assurance guidance for third-party IP," *J. Hardw. Syst. Secur.*, vol. 1, pp. 38–55, Apr. 2017.

[142] J. Heißwolf, "A scalable and adaptive network on chip for many-core architectures," Ph.D. dissertation, Karlsruher Inst. für Technologie (KIT), Karlsruhe, Germany, 2014.
[143] S. Charles and P. Mishra, "A survey of network-on-chip security attacks and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 5, 5 2021.
[144] NXP Semiconductors. (2020). *Security Subsystems for Systems-on-Chip (SoCs)*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.nxp.com/docs/en/white-paper/Security-Subsystems-WP.pdf
[145] N. Fern, I. San, Ç. K. Koç, and K. T. Cheng, "Hiding hardware trojan communication channels in partially specified SoC bus functionality," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 9, pp. 1435–1444, Sep. 2017.
[146] L.-W. Kim and J. D. Villasenor, "A trojan-resistant system-on-chip bus architecture," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2009, pp. 1–6.
[147] L.-W. Kim and J. D. Villasenor, "A system-on-chip bus architecture for thwarting integrated circuit trojan horses," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1921–1926, Oct. 2011.
[148] A. Saeed, A. Ahmadinia, and M. Just, "Secure on-chip communication architecture for reconfigurable multi-core systems," *J. Circuits, Syst. Comput.*, vol. 25, no. 8, Aug. 2016, Art. no. 1650089.
[149] A. Sarihi, A. Patooghy, M. Hasanzadeh, M. Abdelrehim, and A. A. Badawy, "Securing on-chip communications: An on-the-fly encryption architecture for SoCs," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2021, pp. 741–746.
[150] P. Mishra and S. Charles, Eds., *Network-on-Chip Security and Privacy*, 1st ed. Basel, Switzerland: Springer, May 2021.
[151] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 296–310.
[152] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans," in *Cryptographic Hardware and Embedded Systems–(CHES)*, G. Bertoni and J.-S. Coron, Eds. Berlin, Germany: Springer, 2013, pp. 197–214.
[153] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan./Feb. 2010.
[154] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jun. 2008, pp. 15–19.
[155] Trust-Hub. *Trojan Taxonomy*. National Science Foundation. Accessed: Mar. 1, 2024. [Online]. Available: https://www.trust-hub.org/downloads/resource/pdf/Taxonomy.pdf
[156] S. Mulhem, F. Muuss, C. Ewert, R. Buchty, and M. Berekovic, "ML-based Trojan classification: Repercussions of toxic boundary nets," *IEEE Embedded Syst. Lett.*, early access, pp. 1–4, Dec. 4, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10341539
[157] Trust-Hub. *Welcome to Trust-Hub*. National Science Foundation. Accessed: Mar. 1, 2024. [Online]. Available: https://www.trust-hub.org/
[158] S. Adee, "The hunt for the kill switch," *IEEE Spectr.*, vol. 45, no. 5, pp. 34–39, May 2008.
[159] H. Salmani, "COTD: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 338–350, Feb. 2017.
[160] T. Hoque, S. Narasimhan, X. Wang, S. Mal-Sarkar, and S. Bhunia, "Golden-free hardware Trojan detection with high sensitivity under process noise," *J. Electron. Test.*, vol. 33, no. 1, pp. 107–124, Dec. 2016.
[161] J. Zhang and Q. Xu, "On hardware trojan design and implementation at register-transfer level," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, Jun. 2013, pp. 107–112.
[162] Z. Huang, Q. Wang, Y. Chen, and X. Jiang, "A survey on machine learning against hardware trojan attacks: Recent advances and challenges," *IEEE Access*, vol. 8, pp. 10796–10826, 2020.
[163] K. Shuler, *Fundamentals of Semiconductor ISO 26262 Certification: People, Process and Product*. Campbell, CA, USA: Arteris, 2018.
[164] R. Beresnevicius. (May 2019). *Automation in the Aviation Industry—The Future is Automated*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.aerotime.aero/articles/23162-automation-aviation-industry
[165] P. Vadgaonkar and U. Janardhan, "DO-254/ED-80—An application guidelines to redesign/re-engineering airborne electronic hardware," in *Proc. SAE Aerosp. Syst. Technol. Conf.*, Sep. 2016, pp. 2–3.
[166] Synopsys. *What is ASIL?* Accessed: Mar. 1, 2024. [Online]. Available: https://www.synopsys.com/automotive/what-is-asil.html
[167] S. Chonnad, R. Iacob, and V. Litovtchenko, "A quantitative approach to SoC functional safety analysis," in *Proc. 31st IEEE Int. Syst.–Chip Conf. (SOCC)*, Sep. 2018, pp. 197–202.
[168] A. Reschka, *Safety Concept for Autonomous Vehicles*. Berlin, Germany: Springer, 2016, pp. 473–496.
[169] World Nuclear Association. (Jul. Aug. 2023). *Fukushima Daiichi Accident*. Accessed: Mar. 1, 2024. [Online]. Available: https://world-nuclear.org/information-library/safety-and-security/safety-of-plants/fukushima-daiichi-accident.aspx
[170] K. Kurokawa. (2012). *The Official Report of the Fukushima Nuclear Accident Independent Investigation Commission*. Tokyo, Japan. Accessed: Mar. 1, 2024. [Online]. Available: https://www.nirs.org/wp-content/uploads/fukushima/naiic_report.pdf
[171] Z. Xu and J. Abraham, "Design of a safe convolutional neural network accelerator," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2019, pp. 247–252.
[172] A. Nardi, S. Camdzic, A. Armato, and F. Lertora, "Design-for-safety for automotive IC design: Challenges and opportunities," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2019, pp. 1–8.
[173] Synopsys. *Safety Specification Format (SSF)—Automate End-to-End Traceability, Implementation and Verification of Automotive SoC Design*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.synopsys.com/automotive/safety-specification-format.html
[174] Accellera. *Functional Safety Working Group*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.accellera.org/activities/working-groups/functional-safety
[175] AF MODE, *Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, U.S. Department of Defense, Standard MIL-STD-1629A, 1980.
[176] MITRE. (Sep. 2014). *Common Weakness Scoring System (CWSS)*. Accessed: Mar. 1, 2024. [Online]. Available: https://cwe.mitre.org/cwss/cwss_v1.0.1.html
[177] *Common Weakness Risk Analysis Framework (CWRAF)*. MITRE. Accessed: Mar. 1, 2024. [Online]. Available: https://cwe.mitre.org/cwraf/index.html
[178] NIST. *National Vulnerabilities Database—Vulnerability Metrics*. National Institute of Standards and Technologies. Accessed: Mar. 1, 2024. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss
[179] Forum of Incident Response and Security Teams. *Common Vulnerability Scoring System Version 3.0 Calculator*. FIRST. Accessed: Mar. 1, 2024. [Online]. Available: https://www.first.org/cvss/calculator/3.0
[180] J. Geib, B. Santos, D. Berry, M. Baldwin, and B. Kess. (May 2022). *Microsoft Threat Modeling Tool Threats: STRIDE Model*. Accessed: Mar. 1, 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats
[181] EC-Council. *DREAD Threat Modeling: An Introduction to Qualitative Risk Analysis*. Accessed: Mar. 1, 2024. [Online]. Available: https://www.eccouncil.org/cybersecurity-exchange/threat-intelligence/dread-threat-modeling-intro/
[182] Common Criteria, *Common Criteria:2022 Release 1*, Common Criteria Recognition Arrangement (CCRA), 2022. [Online]. Available: https://www.commoncriteriaportal.org/index.cfm
[183] (Nov. 2022). *CEM:2022 Revision 1Common Methodology for Information Technology Security Evaluation*, Common Criteria Recognition Arrangement (CCRA). Accessed: Mar. 1, 2024. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CEM2022R1.pdf
[184] R. Fulton and R. Vandermolen, *Airborne Electronic Hardware Design Assurance*. Boca Raton, FL, USA: CRC Press, 2017.

ENKELE RAMA received the B.S. degree in electrical engineering from The George Washington University, Washington, DC, USA, in 2011, and the M.S. degree in electrical engineering from Michigan State University, East Lansing, MI, USA, in 2014. He is currently pursuing the Ph.D. degree with the Institute for Integrated Systems, University of the Bundeswehr Munich, Germany. From 2015 to 2020, he was with the University of Pristina, Kosovo. His research interests include the safety and security of integrated circuits, hardware Trojans, and the trustworthiness of electronics.

MOUADH AYACHE received the M.Sc. degree in electrical engineering from the Technical University of Braunschweig, Germany. He is currently pursuing the Ph.D. degree with the Institute of Computer Engineering, University of Lübeck, Germany. He is also with Synopsys GmbH, Germany, on EDA methodologies for functional safety and security. His research interests include EDA, IC design and verification, dependable computing, trustworthiness, and the development of ICs for security and safety-critical applications.

RAINER BUCHTY received the Diploma and Ph.D. degrees in computer science from TU Munich, Germany, in 1997 and 2002, respectively. In 2019, he joined the University of Lübeck, Germany, as the Project and Research Manager, with a focus on processor and systems design with Prof. Berekovic. His past experiences include academic positions at KIT, the University of Jena, and TU Braunschweig, including its associated technology transfer company iTUBS GmbH, a temporary professorship at the University of Tübingen, as well as industrial experiences at AMD GmbH Munich and Bell Labs, Holmdel, NJ, USA, and their respective spin-offs Vantis GmbH and Agere Systems.

BERNHARD BAUER received the Ph.D. degree in semiconductor laser amplifiers from Universität der Bundeswehr, Helmut Schmidt, Hamburg, Germany, in 1995. As a Postdoctoral Researcher, he worked on modeling techniques for huge technical systems, followed by ten years of developing telecommunication cross-connects at Siemens. Since 2008, he has been working on safety topics in the areas of avionics, power plants, and automotive. He researches the trustworthiness and dependability of products covering the topics of safety of the intended functionality, functional safety, cybersecurity, and their interference.

MATTHIAS KORB (Senior Member, IEEE) received the master's and Ph.D. degrees in electrical engineering from Rheinisch-Westfälische Technische Hochschule Aachen University, Germany, in 2006 and 2012, respectively. From 2012 to 2016, he was a part of the Research and Development Group DSP Microelectronics, Office of the CTO, Broadcom Corporation, Irvine, CA, USA. In 2016, he joined the Integrated Systems Laboratory, Swiss Federal Institute of Technology, Zürich, as a Senior Researcher. In 2020, he was appointed as a Professor of secure digital circuits with the University of the Bundeswehr Munich. He is also a Senior Design Engineer with Advanced Circuit Pursuit AG, Zürich. His research interests include hardware-efficient VLSI design of building blocks for communication systems especially channel decoders and the application of high-level synthesis to increase productivity in VLSI design.

MLADEN BEREKOVIC (Member, IEEE) received the Ph.D. degree in circuit design for signal processing systems from the Leibniz University of Hannover, Germany. After the Ph.D. studies, he worked on processor design at IBM and led research teams in reconfigurable computing and mobile system design at the Interuniversity Microelectronics Centre (IMEC), Belgium. He is currently the Director of the Institute for Computer Engineering, University of Lübeck, Germany. Before that, he was the Chair of the Computer Engineering Group and the Intel Chair of VLSI design both at the Technical University of Braunschweig (TUBS), Germany. His research interests include circuit and system design for safe, reliable, and secure autonomous systems.

SALEH MULHEM received the Ph.D. degree in cryptographic engineering from the Technical University of Braunschweig (TUBS), Germany. In 2015, he was a member (Ph.D. student) with the Reliability and Security Research Group, Institute of Computer and Network Engineering, TUBS. Since 2020, he has been leading the Trustworthiness Research Group, Institute of Computer Engineering, University of Lübeck, Germany. His research interests include trustworthy electronics, secure SoC design, trustworthy hardware accelerators, and embedded system security. During the Ph.D. studies, he received a full research grant from German Academic Exchange Service (DAAD).