Understanding multidimensional verification: Where functional meets non-functional

Microprocessors and Microsystems 71 (2019) 102867

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Understanding multidimensional verification: Where functional meets non-functional

Xinhui Lai[a,*], Aneesh Balakrishnan[a,b], Thomas Lange[b,c], Maksim Jenihhin[a], Tara Ghasempouri[a], Jaan Raik[a], Dan Alexandrescu[b]

[a] Department of Computer Systems, Tallinn University of Technology, Akadeemia 15A, Tallinn 12618, Estonia
[b] IROC Technologies, 2 Square Roger Genin, 5th floor, Grenoble, 38000, France
[c] Dipartimento di Informatica e Automatica, Politecnico di Torino, Turin, Italy

ARTICLE INFO

Article history:
Received 25 February 2019
Revised 28 June 2019
Accepted 5 August 2019
Available online 5 August 2019

Keywords:
Extra-functional verification
Functional verification
Survey
Taxonomy
Security verification
Reliability verification
Power verification
Machine learning

ABSTRACT

Advancements in electronic systems' design have a notable impact on design verification technologies. The recent paradigms of Internet-of-Things (IoT) and Cyber-Physical Systems (CPS) assume devices immersed in physical environments, significantly constrained in resources and expected to provide levels of security, privacy, reliability, performance and low-power features. In recent years, numerous extra-functional aspects of electronic systems were brought to the front and imply verification of hardware design models in multidimensional space along with the functional concerns of the target system. However, different from the software domain such a holistic approach remains underdeveloped. The contributions of this paper are a taxonomy for multidimensional hardware verification aspects, a state-of-the-art survey of related research works and trends enabling the multidimensional verification concept. Further, an initial approach to perform multidimensional verification based on machine learning techniques is evaluated. The importance and challenge of performing multidimensional verification is illustrated by an example case study.

1. Introduction

Recently, several prominent trends in electronic systems design can be observed. Safety-critical applications in the automotive domain set stringent requirements for electronics certification, the Internet-of-Things (IoT) and Cyber-Physical Systems (CPS) devices are immersed in physical environments, significantly constrained in resources and expected to provide levels of security and privacy [1], ultra-low power feature or high performance. Very complex electronic systems, including those built from the non-certified for reliability commercial-off-the-shelf components, are used for safety- and business-critical applications. These trends along with gigascale integration at nanoscale technology nodes and multi-/many-processor based systems-on-chip architectures have ultimately brought to the front various *extra-functional aspects* of the electronic systems' design at the chip design level. The latter include security, reliability, timing, power consumption, etc. There exist numerous threats causing an electronic system to violate its specification. In the hardware part, these are design errors (bugs), manufacturing defects and variations, reliability issues, such as soft errors and aging faults, or malicious faults, such as security attacks. Withal, there can also be bugs in the software part.

Hardware design model verification detects *design errors* affecting *functional* and *extra-functional* (interchangeably referred as *non-functional*) aspects of the target electronic system. Strictly, the sole *task of extra-functional verification* of a design model is limited to detecting deviations that cause violation of extra-functional requirements. In practice, it often intersects with the task of functional verification [2,14], thus establishing *a multidimensional space for verification*. A "grey area" in distinction between functional and extra-functional requirements may appear when an extra-functional requirement is a part of design's main functionality. E.g., security requirements for some HW design can be split into extra-functional and functional sets if the design's purpose and specified functionality is a system's security aspect, e.g. it is a secure cryptoprocessor.

The contributions of this paper are a taxonomy for multidimensional hardware verification aspects, a state-of-the-art survey of related research works towards enabling the multidimensional verification concept. Further, an approach is evaluated which performs multidimensional verification by using machine learn-

* Corresponding author.
E-mail address: xinhui.lai@taltech.ee (X. Lai).

https://doi.org/10.1016/j.micpro.2019.102867

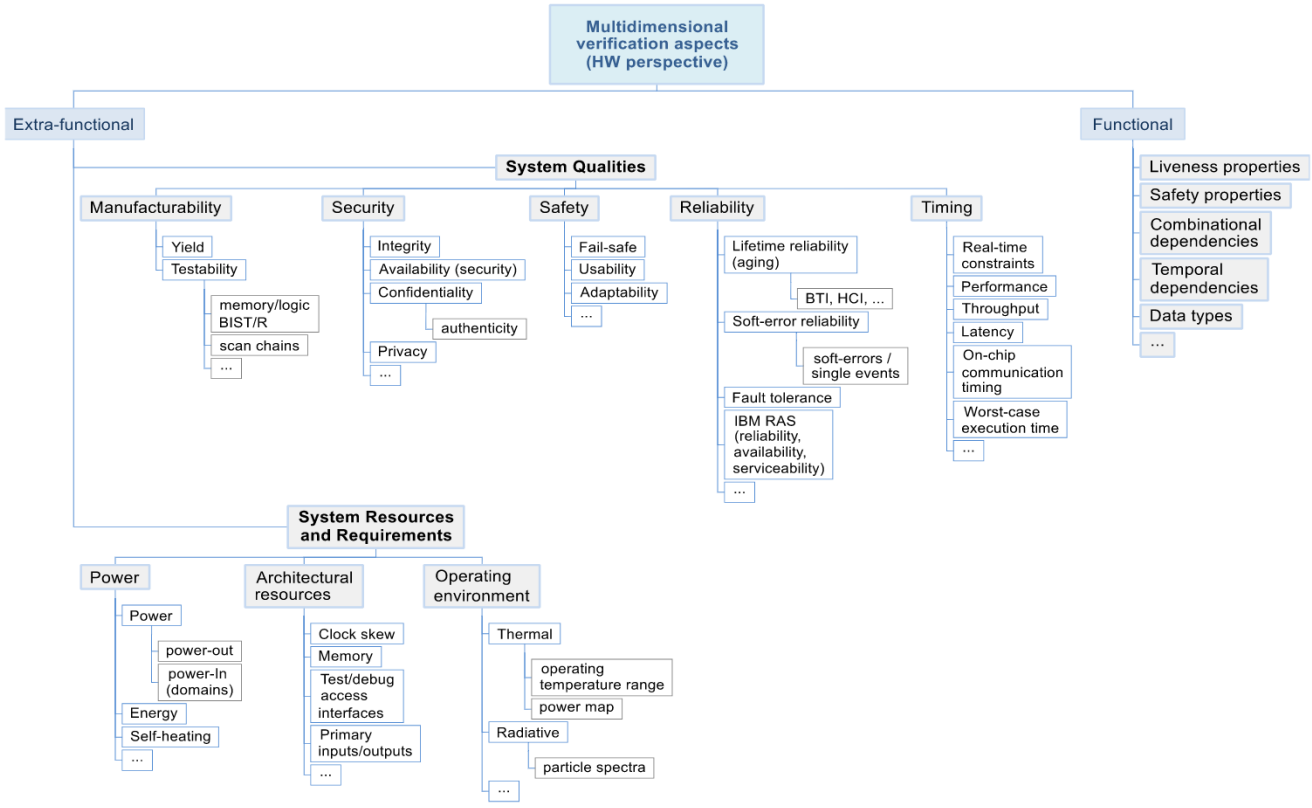

Fig. 1. Taxonomy of multidimensional verification aspects.

ing techniques. The rest of this paper is organized as follows. Section 2 provides a taxonomy of multidimensional verification aspects. Sections 3 proposes a state-of-the-art survey with the key trends in verification for the main extra-functional aspects. Section 4 discusses the multidimensional verification challenges and presents a motivational example for the functional and power verification dimensions. Section 5 proposes adoption of machine learning techniques for support of design's multi-aspect features extraction and verification. Finally, Section 6 draws the conclusions.

2. Taxonomy of multidimensional verification aspects

In practice, relevance of each functional and extra-functional aspect strongly depends on the design type, target system application and specific user requirements. Following the design paradigm shifts, a number of extra-functional aspects have recently received significant academic research attention e.g., security. At the same time, there already exist established industrial practices for measuring and maintaining particular design qualities, e.g. the RAS (Reliability-Availability-Serviceability) aspect introduced by IBM [6]. While in the software engineering discipline, the taxonomy of extra-functional requirements has a comprehensive coverage by the literature [7–12], it cannot be directly re-used for the HW verification discipline because of significant difference in the design models.

Fig 1 introduces a taxonomy of multidimensional verification aspects derived from the performed literature review. The conventional *functional concerns* are *safety and liveness properties, combinational and temporal dependencies* along with *data types*, however this list can be extended for particular designs. The extra-functional aspects can be strictly categorized into two groups: *System Qualities* and *System Resources and Requirements* (in bold). The main system qualities for extra-functional verification are manufacturability of the design, security, in-field safety, reliability during the operational lifespan and a set of timing aspects. The second group embraces the power and architectural resources as well as design constraints set by the operational environment.

Several extra-functional aspects such as *manufacturability*, i.e. primarily *yield* and *testability* against manufacturing defects, *fault-tolerance,* reliability (subject to transient, intermittent and permanent hardware faults) and several aspects from the *System Resources and Requirements* group do not have a direct correspondence in the software engineering discipline because of the distinct nature of faults and specification violations. Other aspects such as *real-time constraints* are very similar between the two domains.

3. Trends in extra-functional verification

Table 1 presents a survey of recent publications targeting extra-functional and multidimensional verification. Here, along with the specific extra-functional aspects details about the design model and verification approach are outlined, i.e., the design under verification type, verification engine, the level of abstraction, design representation language, compute model and the tool operated in the research. We pointed out such key points for all the recent up to 10-year old studies in this area. Further, in the following subsections, we focus on understanding trends for the extra-functional aspects that have the strongest attention in the literature, i.e. security, in-field reliability, timing and power.

3.1. Security aspects

Security is difficult to quantify as today there are no commonly agreed metrics for this purpose [1]. The key targeted security services [16] commonly represented as extra-functional aspects for verification are *confidentiality, integrity* and *availability.* Verifying

Table 1
Survey of the state-of-the-art solutions for extra-functional and multidimensional verification.

Pub.	Year[a]	Extra-functional aspect[b]					Design under verification	Verification engine	Abst. level[e]	Design representation language	Compute model	Tool
		Security	Reliability[c]	Timing[d]	Power	Other aspects						
[19]	2009	confidentiality, integrity	–	–	–	–	HW/SW system	formal, correct-by-construction	SL	AADL	–	OSATE
[20]	2018	confidentiality	–	–	–	–	NoC	unbounded model-checking	RTL	VHDL/Verilog, PSL	LTL	–
[21]	2016	integrity, confidentiality	○	–	–	–	NoC	simulation, HW monitors	RTL	VHDL/Verilog	–	–
[22]	2014	integrity	○	–	–	–	NoC	formal	GL	VHDL/Verilog	–	SurfNoC
[23]	2017	integrity, confidentiality	–	–	–	–	RSN	model check	RTL	ICL	Craig interpolation	CIP solver
[24]	2015	integrity	–	–	–	–	SoC	simulation	RTL	VHDL/Verilog	–	–
[25]	2016	integrity, confidentiality	–	–	–	–	ALU	equivalence check	GL	–	QBF-SAT	–
[26]	2017	integrity	–	–	–	–	SoC	semiformal	GL	–	–	JasperGold SPV
[27]	2016	confidentiality	–	–	–	–	RSN	model check	RTL	ICL	Craig interpolation	CIP Solver
[28]	2017	confidentiality	–	–	–	–	control systems	formal	SL	ASLan++	–	CL-AtSe
[29]	2017	integrity	–	–	–	–	IP cores	semiformal	GL	VHDL	–	mini-SAT
[30]	2015	integrity	–	–	–	–	ISA, pipeline	model check	RTL	–	CTL, LTL	nuXmv SMV
[31]	2018	confidentiality	–	–	–	–	cache	model check	SL	–	CTL	–
[32]	2014	confidentiality	–	–	–	–	cache	model check	RTL/SL	–	FSM	Murphi
[33]	2017	confidentiality	–	–	–	–	cache	model check	RTL/SL	–	FSM	CacheAudit
[34,35]	2013	integrity, confidentiality	–	–	–	–	IPs and SoCs	formal	RTL, GL	Verilog	–	JasperGold SPV
[36]	2018	•	•	–	–	–	MPSoC	model check	SL, RTL	–	Timed Automata	UPPAAL
[41]	2017	–	•	–	–	–	CPS	model check	SL	AADL	Timed Automata	UPPAAL
[42]	2015	–	SER	–	–	–	IP cores	formal	GL/RTL	LDDL	LDDL	Coq
[43]	2010	–	SER	–	–	availability, serviceability	processor	fault inject.	GL	Verilog	–	IBM in-house
[44]	2016	–	•	–	–	availability, serviceability	SoC	fault inject.	RTL	–	–	–
[45,46]	2016	–	○ (LTR)	–	○	thermal	Smart Systems	simulation	SL	IP-XACT, SystemC-AMS	–	–
[47]	2018	–	•	–	–	–	CPS	formal /simulation, HW monitors	RTL	VHDL/Verilog	multiple	multiple

(*continued on next page*)

Table 1 (*continued*)

Pub.	Year[a]	Extra-functional aspect[b]					Design under verification	Verification engine	Abst. level[e]	Design representation language	Compute model	Tool
		Security	Reliability[c]	Timing[d]	Power	Other aspects						
[48]	2014	–	SER	–	–	–	IPs	SAT solver	RTL	VHDL	–	–
[49]	2010	–	SER	–	–	–	IPs, processor	simulation	RTL	VHDL/Verilog	–	–
[50]	2014	–	SER	–	–	–	memory	circuit-level simulation	circuit level	–	–	INFORMER
[51,52]	2018	–	–	comm. constraint	○	–	NoC	fault inject.	RTL	VHDL	–	QoSinNoC
[53]	2011	–	–	RT	–	–	memory	model check	RTL	REAL/AADL	–	Ocarina
[54]	2010	–	–	RT	–	–	Scheduler of RT system	model check	–	Promela	Time Petri-net	SPIN
[55]	2010	–	–	latency	–	–	RT emb. system	model check	SL	AADL	–	YICES
[56]	2017	–	–	performance	○	–	NoC, HW/SW architectures	simulation	SL	Graph Assembly Language	connectivity graphs	ArchOn
[58]	2012				●	–	IPs	simulation	SL	SystemC	–	–
[59]	2016				●	–	DSP cores	simulation	SL,GL, RTL	SystemC	–	Powersim
[62]	2017	–	–	RT	○	–	automotive CPS	model check	SL	C, EAST-ADL	Timed Automata	UPPAALsdv
[63]	2016	–	–	–	●	–	IPs	Semiformal ABV	RTL	VHDL/Verilog; SystemC	Hidden Markov Model	–
[64]	2012	–	–	execution time	○	–	distributed emb. system	simulation	SL	SystemC	–	–
[65]	2016	–	–	performance RT	○	thermal	HW/SW platform	semiformal	RTL,TLM,SL	UML,C++,VHDL SystemC-AMS	HIF	HIFSuite
[66]	2009	–	–	throughput	–	–	SoC/FPGA	simulation	RTL	Verilog/VHDL	–	Modelsim
[67]	2018	–	–	throughput	–	–	NoC	simulation	RTL	System Verilog	–	UVM
[68]	2014	–	–	–	–	connectivity	SoC	symbolic model checking	RTL, TLM	Verilog	–	Incisive Formal Verifier
[70]	2016	–	–	–	–	memory consistency	processor	simulation	ISA	ruby	–	McVerSi
[73]	2011	–	–	–	●	thermal	SoC	simulation	SL,GL, RTL	SystemC	–	Power-Mixer, -Depot, -Brick
[74]	2015	–	–	–	●	–		simulation	SL,TLM	SytemC	–	Power Kernel Tool
[75]	2011	–	–	–	●	–	SoC	simulation	SL	SystemC	–	Powersim

[a] *only conference, journal and industrial white papers published in the last 10 years were selected for this survey.*
[b] *● – this aspect is the main focus in the paper; ○ – this aspect is partially addressed.*
[c] *LTR – lifetime reliability; SER – soft-error reliability;.*
[d] *RT – real-time constraints;.*
[e] *GL – gate level; SL – system level; ISA – instruction set architecture level; TLM – transaction level model.*

security aspects is highly dependent on the type of attack and the attacker model assumed.

Many of the existing works in security verification (e.g. [22,24,26,29,30]) are focusing on the integrity attribute, mostly addressing hardware trojan detection. There also exist works that additionally target [19,21,23,25,34] or are exclusively considering [27,28] the confidentiality aspect. Several solutions in security verification are restricted to specific target architectures or types of modules such as Reconfigurable Scan Networks (RSNs) [23,27] or macro-asynchronous micro-synchronous pipelines [30]. To that end, for complex hardware architectures (e.g. large IEEE1687 Reconfigurable Scan Networks or MPSoCs) the specific on-chip security features to be verified also tend to be very sophisticated. These may include on-chip mechanisms for attack prevention (firewalls, user management, communications' isolation), attack protection (traffic scrambling, encryption) and attack resilience (checkers for side-channel attacks, covert channel detection, attack recovery mechanisms). Several works consider security verification for NoC-based MPSoCs. [20] proposes a method to formally verify the correctness and the security properties of a NoC router. Some solutions in the security verification of NoCs do indirectly address reliability due to the fact that they implement hardware monitors that allow avoiding both, attacks and in-field faults [21,22].

According to recent surveys [37] and [38] cache access driven side-channel attacks have become a major concern in hardware security. In modern processors, deep hierarchy of cache memory is implemented to increase system performance. However, this makes modern computing systems, including IoT devices, vulnerable to cache side-channel attacks. There exist several works addressing verification of the cache security. In [31], the authors propose.

Computation Tree Logic (CTL) based modeling of timing-driven and access-driven cache attacks. This work concentrates on formally describing the attack types. Zhang and Lee [32] models cache as a state machine and proposes a metric based on the non-interference condition to evaluate the access-based cache vulnerability. Canones et al. [33] proposes a model to formally analyze the security of different cache replacement policies. None of the above-mentioned works consider multiple dimensions, or aspects.

An approach that is designed for modeling a multitude of extra-functional aspects is the model-based engineering example of Architecture Analysis and Design Language (AADL) [19]. While, in principle, AADL allows representing several extra-functional aspects (called quality attributes in AADL), Hansson et al. [19] only concentrates on analysis of confidentiality as a part of verifying security in a system with multiple levels of security. The authors in [36] have targeted a general Uppaal Timed Automata based multi-view hardware modeling and verification approach taking into consideration of the security view. The survey of related literature clearly shows that, up to this moment, there is virtually no work considering security verification in combination with other extra-functional aspects.

3.2. Reliability aspects

The key drivers for the reliability aspect in today's designs are the recent industrial standards in different application domains such as IEC61508, ISO26262, IEC61511, IEC62279, IEC62061, RTCA/DO-254, IEC60601, etc. Integrated circuits used in high-reliability applications, e.g. complying with high (Automotive) Safety Integrity Level - (A)SIL, must demonstrate low failure rates (modelled by FIT – Failures in Time) and high fault coverage (e.g. Single-Point Failure Metric SPFM and Latent Fault Metric LFM). These requirements ultimately mandate extra-functional validation efforts for reliability analysis, such as Failure Mode and Effects (Criticality) Analysis - FME(C)A and imply generalized use of methods and features, such as safety mechanisms, for error management. *Functional safety* is a property of the complete system rather than just a component property because it depends on the integrated operation of all sensors, actuators, control devices, and other integrated units. The goal is to reduce the residual risk associated with a functional failure of the target system below a threshold given by the assessment of severity, exposure, and controllability.

The dominant threats for reliability are, first, random hardware faults such as transient faults by radiation-induced single event effects or soft errors [15], i.e. a subject for *Soft-Error Reliability* (SER). Second, these are extreme operating conditions, electronic interference and intermittent to permanent faults by process or time-dependent variations, such as aging induced by Bias Temperature Instability (BTI) [13], where the latter is a subject for *Life-Time Reliability* (LTR). Reliability verification challenge is emphasized by the adoption of advanced nanoscale implementation technology nodes and high complexity of systems, utilizing tens or hundreds of complex microelectronic components and embedding large quantities of standard logic and memory. Moreover, these designs integrate IP cores from multiple design teams making reliability evaluation task to be scattered and complex. Initiatives such as RIIF (Reliability Information Interchange Format) [39], allow the formalization, specification and modeling of extra-functional, reliability properties for technology, circuits and systems.

Similar to other aspects, reliability in large complex electronic systems, e.g. safety-critical CPSs, may be tackled starting at high level of abstraction. System's fault tolerance is formally checked using UPPAAL and timed automata models generated from AADL specifications [41]. HW design models and tools at such a level also enable verification of interference of several extra-functional design aspects [36]. There are research works relying on design soft-error reliability verification by fault-injection campaigns, e.g. [49], or formal analysis, e.g. error-correction code (ECC) based mechanisms against single-bit errors in memory elements [48]. Burlyaev and Fradet [42] proposes a general approach to verify gate-level design transformations for reliability against single-event transients by soft errors that combines formal reasoning on execution traces. Thompto and Hoppe [43] and Kan et al. [44] focus on the RAS (Reliability, Availability and Serviceability) group of extra-functional aspects outlined by IBM for complex processor designs where embedded error protection mechanisms and designs intrinsic immunity (due to various masking) to errors is evaluated by fault injection. Vinco et al. [45,46] propose extensions to system descriptions in the IP-EXACT format to enable multi-layer representation and simulation of several mutually influencing extra-functional aspects of smart system designs such as lifetime reliability, power and temperature. A complex approach to verification of multiple reliability concerns (soft errors, BTI, etc.) across layers in industrial CPS designs is proposed in [47] as a collaborative research result in the IMMORTAL project. Last but not least, addressing the need for reliability verification automation tools, in [50], authors propose a fully automated tool INFORMER to estimate memory reliability metrics by circuit-level simulations of failure mechanisms such as soft-errors and parametric failures.

The survey clearly shows that currently there is a very small number of works considering verification of reliability together with other aspects.

3.3. Timing aspects

Functional temporal properties are essential part of sequential designs' specification that are often modelled for functional verification by Computational Tree Logic (CTL), applied for formal approaches, and Linear Temporal Logic (LTL) temporal assertions expressed arbitrarily, e.g. in Property Specification Language (PSL), System Verilog Assertions (SVA) or systematically, e.g. in Universal

Verification Methodology (UVM). In the extra-functional context, these can be extended to specific requirements and properties such as: *real-time (RT), performance, throughput, latency, on-chip communication time constraint, worst-case execution time* constraints, etc. Several works have been widely studying these timing properties. Some researchers are mainly focused on generating timing properties to reduce the verification effort, for example, state space and cost [54,56,65]. Other works instead use the timing properties to assess whether the system under verification is correctly functioning or not [55,62,64]. In the following, we discuss state of the art for each timing aspect.

A real-time system describes hardware and software systems subject to a *real-time constraint*, that ensures response within a specified time. The correctness of the function depends both on the correctness of the result and also the timeliness of the periods. In [54], an approach to verify the timed Petri-Net model is proposed. A non-instantaneous model is abstracted from the timed Petri-Net model in a hierarchical structure. The non-instantaneous model which is verified with a model-checking tool is used to reduce the state space of the timed Petri-Net model for verification with a satisfiability modulo theories solvers [76,77]. The timed Petri-Net is used to model the interacting relations of the software components and the binding relations between software and hardware in a certain period of time. Görgen et al. [65] introduces a tool called CONTREX to complement current activities in the area of predictable computing platforms and segregation mechanisms with techniques to compute real-time properties. CONTREX enables energy-efficient and cost-aware design through analysis and optimization of real-time constraint. The authors in [62] proposed a method to combine real-rime constraint aspect of a model with energy-aware real-time (ERT) behaviors of the model into UPPAAL for formal verification.

Throughput is a measure of how many units of information a system can process in a given amount of time. In [66], a verification environment has been proposed to estimate the throughput of a SoC. The intention of the paper is to judge whether the verification system can handle SOC verification and provide the necessary performance in terms of speed and throughput. Khamis et al. [67] introduced a Universal Verification Methodology (UVM) environment to measure throughput of a NoC. UVM is a SystemVerilog class library explicitly designed to help and build modular reusable verification components and test-benches. It is an industry standard, so it is possible to acquire UVM IP from other sources and reuse them.

Performance refers to the amount of work which is done during a process, for instance, executing instructions per second. In [56], a framework has been developed to analyze performance of a system design. The framework is based on stochastic modeling and simulation and it is applied on a set of NoC topologies. The methodology uses a selective abstraction concept to reduce complexity.

When referring to hardware, *latency* is the time required for a hardware component to respond to a request made by another component. However, in the cast of hardware, latency is sometimes referred to as the *access time*. In [55], an analysis tool is developed to work with the AADL models [78] to assure the correctness of a scheduling model that binds the relation of different components in a model.

On-chip communication time constraints refer to the requirements on the start and end times of each task in a system critical path, which is the sequence of tasks that cannot be delayed without delaying the entire system. For instance, in [51] and [52] a framework has been proposed, which is based on a set of quality of service aware NoC architectures along with the analysis methodology including selected relevant metrics that enable an efficient trade-off between guarantees and overheads in mixed-criticality application scenarios. These architectures overcome the notion of strictly divided regions by allowing non-critical communication pass through the critical region, providing they do not utilize common router resources. Such problem formulation is relevant to facilitate the usage of NoC technology by safety-critical industries such as avionics.

The *worst-case execution time* of a computational task is the maximum length of time the task could take to execute on a specific hardware platform. The designer of a system can employ techniques such as schedulability analysis to verify that the system responds fast enough [40]. For instance, Zimmermann et al. [64] presents an approach to generate a virtual execution platform in SystemC to advance the development real-time embedded systems including early validation and verification. These virtual execution platforms allow the execution of embedded software with strict consideration of the underlying hardware platform configuration in order to reduce subsequent development costs and to allow a short time-to-market by tailoring and exploring distributed embedded hardware and software architectures.

Last but not least, a few works also take into account dependencies between several extra-functional aspects. For instance, the work in [62,65] and [56] present the effect of optimizing timing properties (performance and latency) on power consumption or the study in [64] performs the effect of decreasing execution time on power consumption. Such analysis is mostly limited to two extra functional aspects or neglected at all [53–55,69], while design timing constraints can strongly influence not only power consumption but reliability, security, availability, etc. as well as functional properties.

3.4. Power aspects

In commercial flows, verification of the power aspect can be addressed relatively independently from the functional verification dimension. The *power intent* and detailed power modelling can be done starting at TLM or RTL with minimal interference with the HDL functional description, e.g. using the Accellera introduced Unified Power Format (UPF) employed for power-aware design verification automation by commercial tools especially with the latest UPF3.0 [60] or Cadence/Si2 Common Power Format CPF [61]. For the advanced device implementation technologies, power specification implies *multi-voltage design* with up to tens of *power domains* and may consider dynamic and adaptive voltage scaling.

In the recent research works, design verification against the power aspect is performed at different abstraction levels with a trade-off between speed and accuracy. Some works such as [58,59,74,75] perform power analysis at system level targeting high simulation speed and low power optimization flexibility similar to the accuracy achievable at lower levels. In [58], the authors applied their approach to SRAM and AES encryption IPs and obtained a significant simulation speed-up in comparison to gate-level simulation with a high fidelity of the system-level power simulation. A promising software tool for power simulation in SystemC designs is the Powersim framework [59,75]. In [59], a methodology to estimate the dissipation of energy in hardware at any level of abstraction is proposed. In [75], the authors propose a SystemC class library aimed at calculation of energy consumption of hardware described at system level. The work in [73] introduces a series of tools (PowerBrick (construct power library for standard cell library), PowerMixer (for RTL/gate-level estimator), PoweMixer^{ip} (IP-based model builder), PowerDepot (estimate system-level power consumption)) which can be tightly linked and enable the power analysis from layout, gate-, RT-, IP- to system level with a good simulation speed while retaining high accuracy. The power aspect verification could benefit from a holistic multi-level modelling, such as e.g. [17] available for functional verification. Rafiev et al.

[56], Vinco et al. [45,46], Kang et al. [62], Zimmermann et al. [64], Görgen et al. [65], are aimed at methodologies suitable for specific applications (such as cyber-physical system [62]) that assume verification of extra-functional aspects such as power, timing, thermal at the system level.

This extra-functional aspect has a tight relation to the implementation technology assumed for the synthesis of the design model under verification. With planar bulk MOSFET technology known for exponential growth of the static leakage power for smaller device geometries and employment of FinFET and Tri-Gate-Transistors in the advanced technology nodes, the CMOS device parameters are essential for this analysis [57].

3.5. Machine learning based techniques

The complex problem of multidimensional verification can be assisted by the recent advances in the machine learning discipline. This type of approaches (along with e.g. evolutionary algorithms) is particularly suitable for multi-aspect optimization problems where formal deterministic approaches may lack scalability.

Machine Learning (ML) is the concept of a machine learning from examples and making predictions based on its experience, without being explicitly programmed [82]. Previous works have shown that ML can be used for verification purposes at different levels. In [83], machine learning was introduced in physical design analysis. The feasibility of ML in physical design verification (e.g., lithography hotspot detection) was investigated, and a reference model for application was presented. Based on this work [84] used ML to increase the speed of the performance evaluation (power and area) of a circuit design after physical design by a factor of 40 as well as performing a Design Rule Check. In [85], ML was used to predict the timing behavior of the final floorplan of a circuit during the Place & Route routine and thus, shifting the analysis to an earlier design stage. In [79], the analysis is moved even to higher abstraction level. The high-level synthesis (HLS) resource usage and timing estimation was improved by train ML models with data from real implementations. Thus, the design flow can be assisted with machine learning and predict accurate values even in very early design stages. Machine learning was further applied for Security Verification in [80,81,86], where it was used to detect Hardware Trojans based on features from the Gate Level Netlist. In Section 5, we propose an approach to assist the multidimensional verification flow by using machine learning techniques to estimate a reliability metric, as well as timing metric.

4. The challenges of multidimensional verification

The performed analysis of the state of the art has outlined a gap in methodologies and tools for holistic multidimensional verification of hardware design models.

Different from functional verification, approaches for extra-functional hardware design aspects' verification remain underdeveloped even when tackled in isolation. Here, one of the key issues is a lack of established metrics for verification confidence. For a particular functional verification plan, the functional dimension usually includes conventional structural (code) coverage metrics, functional coverage [3] in form of asserted and assumed properties and design parameters along with stimuli quality assessment by model mutations [18]. The metrics for confidence in extra-functional dimension verification results may be challenging as in practice the requirements are *subjective* and can be specified as a mixture of *quantitative* and *qualitative constraints*. Accurate hardware verification in a particular dimension requires both sufficient extra-functional design modeling and the extra-functional aspects target modeling [36]. There is a limited number of dedicated commercial tools and common standards for extra-functional verification flows. In particular, for the security dimension the JasperGold SPV [35] is one of the few such commercial tools that stand out from the academic research frameworks. Finally, the issue of eliciting the extra-functional requirements [4,5] is a challenging task as ambiguity and (sometimes conflicting) interdependency of the extra-functional aspects in the specifications increases complexity and may leave gaps in the multidimensional verification plans.

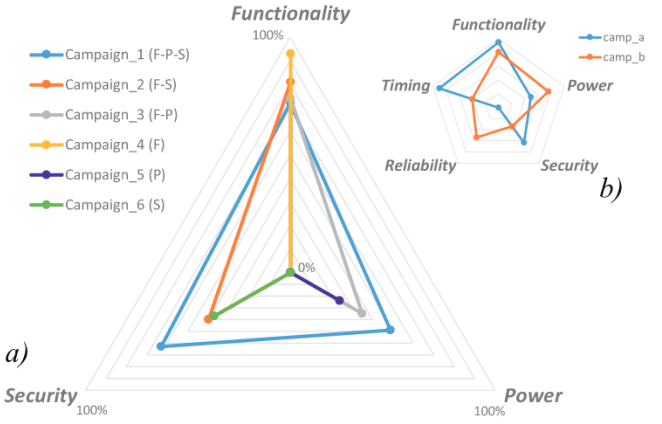

Fig. 2. Multidimensional verification campaigns (Radar-chart n-dimensional visualization).

Unfortunately, there is no established hardware design methodology supporting multidimensional verification plans for mutually influencing functional and extra-functional aspects. There is a very limited number of research works going beyond analysis of one extra-functional verification aspect under constraints of another as the complexity of the problem grows extremely fast with the number of dimensions (interdependent constraints) and the electronic system size. The first works in this direction are, for example, Vinco et al. [46] and Vain et al. [36].

Ultimately, results of multidimensional verification campaigns proposed in this work are subject to be represented in a multi-dimensional space, as illustrated in Fig. 2a. Here is shown an illustration of six hypothetical independent verification campaigns in a three-dimensional verification space. A verification campaign in this example shows the level of confidence in the different dimensions - (F)unctionality, (P)ower and (S)ecurity. In this illustrative example, only three aspects are taken into consideration. Obviously, on the demand the verification engineers can involve different dimensions. Here, the different colors of the lines represent different multi-dimensional spaces e.g. as Campaign_1 in blue lines stand for the verification result considering three extra functional aspects i.e., functional, power and security aspects at the same time. The figure shows the interdependency of these three requirements and thus can help the designers to choose the most suitable design combination. Subsequently, Compaign_2 represents the combination of functional and security aspects, Compaign_3 demonstrates the combination of functional and power aspect, etc. Thus the *Radar-charts, as shown in* Fig. 2b, are an instrument for summarizing multidimensional verification results for a large number of dimensions, (where the dimensions can be ordered to emphasize correlation or interdependencies between adjacent dimensions).

4.1. Motivational example

Single-dimension verification campaigns ignoring interdependencies between the dimensions may lead to gaps in the overall electronic system quality. As an example to show the importance of multidimensional verification, let us consider an actual verification campaign of an open-source NoC framework Bonfire [71,72].

```
process(write_en, write_pointer) begin --write pointer bug
  if write_en = '1' then
  write_pointer_in <= write_pointer(0)&write_pointer(3 downto 1);          -- Bug f1!
  else
    write_pointer_in <= write_pointer;
  end if;
end process;

process(read_en, empty, read_pointer) begin --read pointer bug
  if (read_en = '1' and empty = '0') then
    read_pointer_in <= read_pointer(0)&read_pointer(3 downto 1);           -- Bug f1!
  else
    read_pointer_in <= read_pointer;
  end if;
end process;
```

```
process(write_en, write_pointer )begin --write pointer
  if write_en = '1' then
  write_pointer_in <= write_pointer(2 downto 0)&write_pointer(3);
  else
    write_pointer_in <= write_pointer;
end if;
end process;

process(read_en, empty, read_pointer) begin --read pointer
  if (read_en = '1' and empty = '0') then
    read_pointer_in <= read_pointer(2 downto 0)&read_pointer(3);
  else
    read_pointer_in <= read_pointer;
  end if;
end process;
```

Fig. 3. Bug f1 and its correction.

```
process(Healthy_packet, reset_counters, healthy_counter_out) begin
  if reset_counters  = '1' then
      healthy_counter_in <=  (others => '0');
  elsif Healthy_packet = '1' then                -- Bug p1!
      healthy_counter_in <= healthy_counter_out + 1;
  else
      healthy_counter_in <= healthy_counter_out;
  end if;
end process;
```

```
process(Healthy_packet, reset_counters, healthy_counter_out,  faulty_counter_out) begin
  if reset_counters  = '1' then
      healthy_counter_in <=  (others => '0');
  elsif Healthy_packet = '1' and faulty_counter_out /= std_logic_vector(to_unsigned(0, faulty_counter_out'length)) then
      healthy_counter_in <= healthy_counter_out + 1;
  else
      healthy_counter_in <= healthy_counter_out;
  end if;
end process;
```

Fig. 4. Bug p1 and its correction.

The design under verification is in RTL VHDL and implements a 2×2 NoC infrastructure (processing elements excluded). The verification plan considered 2-dimensional verification campaign targeting *functionality* and *power consumption* requirements. For the former, assertion-based functional verification by simulation was employed targeting statement, branch, condition and toggle coverage metrics and satisfaction of a set of temporal simple-subset PSL assertions. For the latter, a set of power targets were extracted for the targeted silicon implementation assuming a particular switching activity (set to 12 mW in this example).

Among documented design errors in the Bonfire project, the bug *f1*, as shown in Fig. 3, is an example of a functional misbehavior due to improper usage of write and read pointers in the FIFO. The figure represents the code errors in the red line and the corrected versions of the code lines in blue. The bug f1 and the bug p1 demonstrate the error in Figs. 3 and 4, respectively. The bug *p1* causes violations of specified power consumption targets because of unnecessary excessive use of a fault-tolerance structure related counter. The report of such a power consumption is described in Table 2. The power consumption is shown in the cell Total Power which is composed of the dynamic power, i.e. the Switching Power in the interconnects and the Internal Power in the logic cells, and the insignificant (for the target technology) static leakage power Leak Power. As summarized in the first row, for the bug f1 the Total Power is equal to 10.211 (consistence with the power consumption requirement). Similarly, in the third row, which represents the power consumption for the correct version of the code, the total power is equal to 10.184. This report prove that even if there is a bug (bug f1) in the code but still the power consumption requirement is met. In contrary, for the bug p1, even though there is no functional errors, the Total Power consumption is reported which is equal to 22.137. Thus the bug p1 results in a double power consumption compared to the correct implementation and violates the power targets in the specification. This fact prove that it is critical to know how and where the code should be modified in order to reduce the power consumption as well as maintain functional correctness. In general, the above simple motivation example demonstrates the challenge of interdependency of different aspects when requirements in more than one dimension are present.

5. Machine learning to tackle the challenges of multidimensional verification

As it can be seen in the previous sections, multidimensional verification is a complex multi-aspect optimization problem. Machine learning algorithms are known to be able to learn complex relationships and have been used for several optimization problems. Section 3.5 has shown that machine learning techniques were already successfully used for estimating several different single verification metrics. This suggests that machine learning can be also used for solving multidimensional verification problem. There-

Table 2
Power consumption of the Bonfire system implementation: corrected and with bugs f1 and p1.

Bonfire system Implementation	Switching Power (mW)	Internal Power (mW)	Leak Power (pW)	Total Power (mW)
with *f1* bug	0.783	9.427	7.50e+05	10.211
with *p1* bug	0.757	21.379	6.93e+05	22.137
corrected	0.666	9.518	7.43e+05	10.184

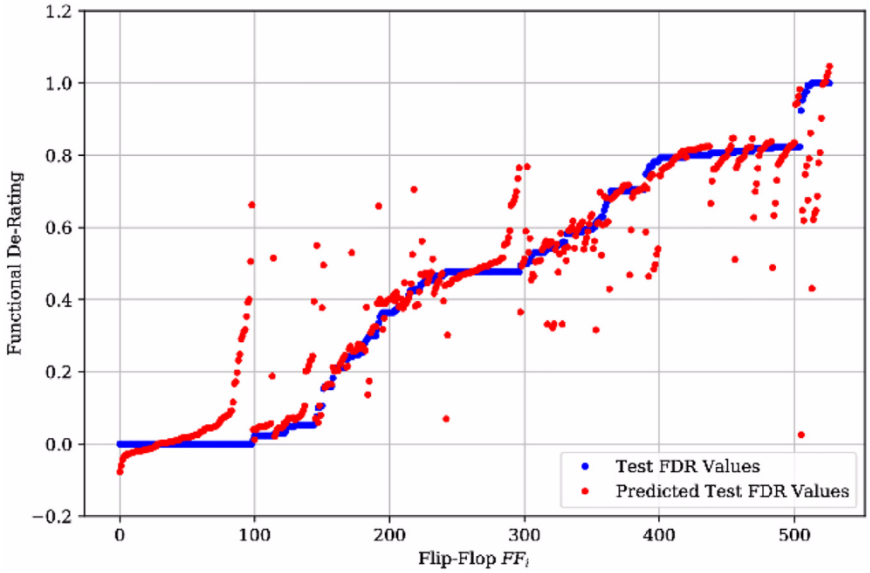

Fig. 5. Prediction of Functional De-Rating factors of the test data set by using a Support Vector Machine regression model (Training Size = 50%, Coefficient of Determination $R^2 = 0.844$).

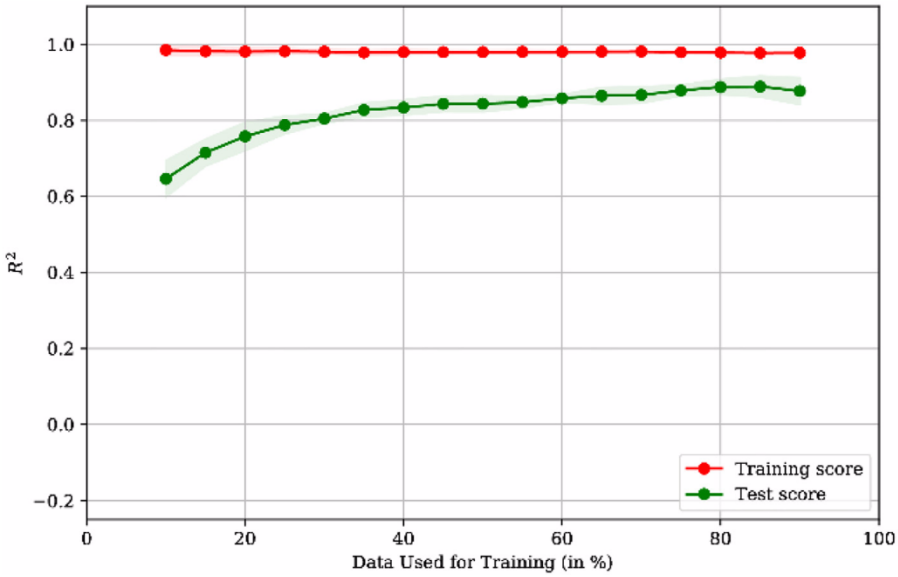

Fig. 6. Learning Curve for the Functional De-Rating prediction by using a Support Vector Machine regression model with different training sizes.

fore, an initial approach is proposed which is based on machine learning techniques in order to tackle this multidimensional verification challenge.

5.1. Proposed methodology

The proposed approach targets to predict two different verification metrics based on the same feature set extracted from the gate-level netlist of a given circuit. These two different metrics are Prediction of De-Rating and Path delay. The first metric to predict is the De-Rating or Vulnerability Factor, which are related to the reliability verification flow and a major metric of the failure analysis. The second metric is the path delay and related to the timing analysis. This metric is usually obtained during the synthesis or place and route stage of the design development. Therefore, machine learning can help to shift the analysis to an earlier design stage.

A possible application scenario consists in extracting a list of circuit feature and training a ML tool with a limited set of reference inputs (the values of the selected circuit features) and

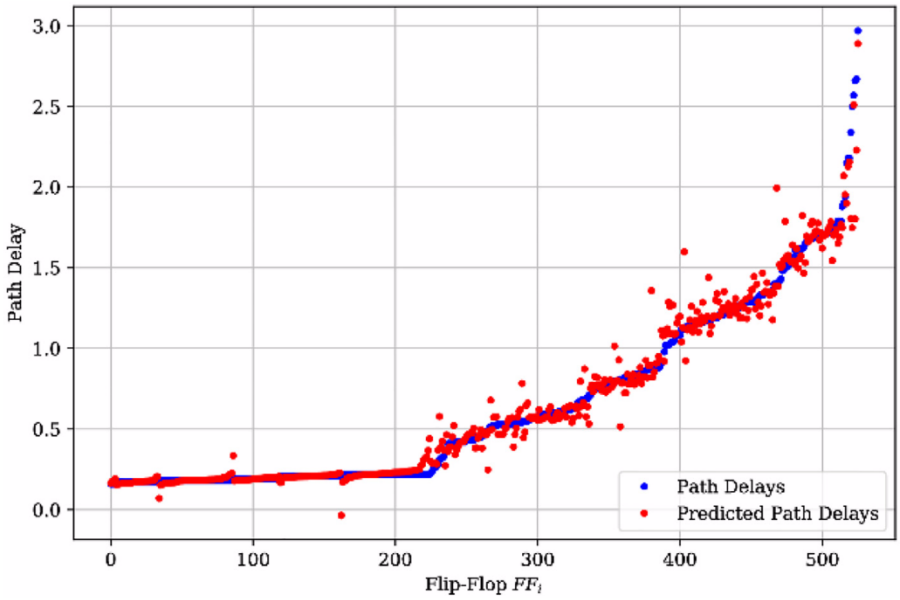

Fig. 7. Prediction of Path Delays of the test data set by using a Support Vector Machine regression model (Training Size = 50%, Coefficient of Determination $R^2 = 0.975$).

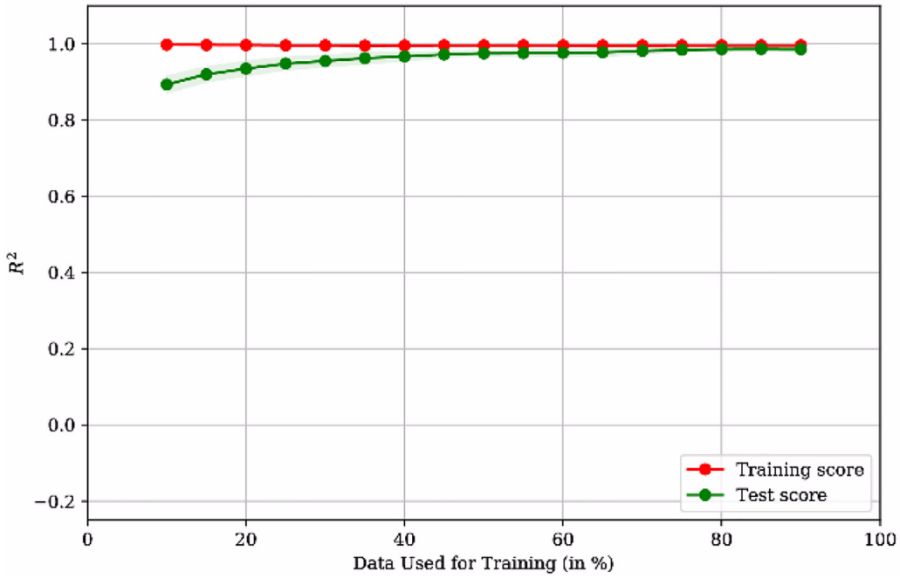

Fig. 8. Learning Curve for the Path Delay prediction by using a Support Vector Machine regression model with different training sizes.

expected outputs (reliability and timing metrics). Depending on the exhaustiveness of the training campaign, the trained ML tool can provide actual reliability metrics from a limited list of circuit features while spending far less resources (CPU time, EDA tools licenses, man-power) than using classical methods.

5.2. Prediction results

The proposed idea was implemented and evaluated on a practical example. Therefore, a set of features is defined which characterizes each flip-flop instance in the circuit. The feature set is composed of static elements (cell properties, circuit structure, synthesis attributes) and dynamic elements (signal activity). After extracting the features for the full list of circuit instances, reference data was obtained. The Functional De-Rating per flip-flop was determined through first-principles fault simulation approaches and the path delay was extracted by a classical static timing analysis. One part of the reference dataset is used to train the machine learning model and the remaining data is used to validate and benchmark the accuracy of the trained tool.

As a circuit under test, the Ethernet 10GE MAC Core was used which is available as RTL description from OpenCores. The circuit consists of control logic, state machines, FIFO controllers and memory interfaces. By synthesizing the design with NanGate FreePDK45 Open Cell Library, 1054 flip-flops have been identified and the corresponding features have been extracted.

Several machine learning models have been evaluated, such as the Linear Least Squares, Ridge (with linear and non-kernels), k-Nearest Neighbors (k-NN), Decision Tree (CART) and Support Vector regression (SVR, with linear and non-linear kernels). It has been noted that especially the linear models are not very suitable to predict the reliability metrics. The non-linear models perform much better and the Support Vector regression with Radial Basis Function (RBF) as kernel functions is among the best. There-

fore, the SVR model with RBF kernel function is used for the following presentation of the prediction results. Figs. 5 and 7 show the prediction of the two metrics. When 50% (527 flip-flops) of the data are used to train the model and the remaining 50% was used to evaluate the model. The performance of regression models is usually evaluated by using the Coefficient of Determination (R^2) score and the model reaches a score of $R^2 = 0.844$ to predict the Functional De-Rating and $R^2 = 0.975$ to predict the path delay. Figs. 6 and 8 show the learning curve of the model. This curve describes the performance of the model for different sizes of the data set used for training and the remaining data set used for the evaluation. The learning curves suggest that by using more than 50% of the available data for training doesn't significantly improve the prediction performance. However, it can also be seen that by using less than 50% still a valuable prediction can be performed. Thus, by allowing a slight reduction of accuracy, the cost of an exhaustive analysis can still be reduced.

The practical example has shown that machine learning can be successfully applied for different verification purposes. In order use ML to support the multidimensional verification problem, features from different design stages need to be extracted and used to train a unified model or several separated models. These can be used to predict the required verification metrics.

6. Conclusion

In the recent years, numerous extra-functional aspects of electronic systems were brought to the front and imply verification of hardware design models in multidimensional space along with the functional concerns of the target system. Targeting at understanding of this new verification paradigm, we have performed a comprehensive analysis of the state of the art and presented a taxonomy for multidimensional hardware verification aspects, an up-to-date survey of related research works and trends towards enabling the multidimensional verification concept and investigated the potential of machine learning based techniques to support the concept. As the result of the performed analysis of the state of the art we have outlined a gap in methodologies and tools for holistic multidimensional verification of hardware design models and the key challenges.

Declaration of competing interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

Acknowledgments

This research was supported in part by projects H2020 MSCA ITN RESCUE funded from the EU H2020 programme under the MSC grant agreement no. 722325, by the Estonian Ministry of Education and Research institutional research grant no. IUT19-1 and by European Union through the European Structural and Regional Development Funds.

References

[1] I. Verbauwhede, Security adds an extra dimension to IC design: future IC design must focus on security in addition to low power and energy, in: IEEE Solid-State Circuits Magazine, 9, Fall 2017, pp. 41–45.
[2] W. Chen, S. Ray, J. Bhadra, M. Abadir, L.C. Wang, Challenges and trends in modern SoC design verification, in: IEEE Design & Test, 34, Oct. 2017, pp. 7–22.
[3] A. Piziali, Functional Verification Coverage Measurement and Analysis, Springer, 2008.
[4] S. Ullah, M. Iqbal, A.M. Khan, A survey on issues in non-functional requirements elicitation, in: Int. Conf. on Computer Networks and Information Technology, Abbottabad, 2011, pp. 333–340.
[5] L.M. Cysneiros, E. Yu, Non-Functional requirements elicitation, in: J.C.S. do Prado Leite, J.H. Doorn (Eds.), Perspectives On Software Requirements. The Springer International Series in Engineering and Computer Science, 753, Springer, Boston, MA, 2004.
[6] M.L. Fair, Reliability, availability, and serviceability (RAS) of the IBM eServer z990, IBM J. Res. Dev. 48 (3.4) (May 2004) 519–534.
[7] L. Chung, B. Nixon, E. Yu, J. Mylopoulos, Non-Functional requirements, Software Engineering, Kluwer Academic, 2000.
[8] P. Singh, A.K. Tripathi, Exploring problems and solutions in estimating testing effort for non functional requirement, Int. J. Comput. Technol. 3 (2b) (2012) 284–290.
[9] E.R. Poort, N. Martens, I. Van de Weerd, H. Van Vliet, How architects see non-functional requirements: beware of modifiability, in: Requirements Engineering: Foundation for Software Quality, Springer, Berlin Heidelberg, 2012, pp. 37–51.
[10] D. Ameller, C. Ayala, J. Cabot, X. Franch, How do software architects consider non-functional requirements: an exploratory study, in: Requirements Engineering Conference (RE), 2012, pp. 41–50.
[11] M. Glinz, On non-functional requirements, in: Requirements Engineering Conference, 2007. RE'07, IEEE, 2007, pp. 21–26.
[12] L. Motus, Analytical study of quantitative timing properties of software, 5th EUROMICRO Workshop on Real-Time Systems, 1993.
[13] M. Jenihhin, G. Squillero, T.S. Copetti, V. Tihhomirov, S. Kostin, M. Gaudesi, F. Vargas, J. Raik, M. Sonza Reorda, L. Bolzani Poehls, R. Ubar, G.C. Medeiros, Identification and rejuvenation of NBTI-Critical logic paths in nanoscale circuits, JETTA 32 (3) (June 2016) 273–289.
[14] J. Bhadra, M.S. Abadir, L.C. Wang, S. Ray, A survey of hybrid techniques for functional verification, in: IEEE Design & Test of Computers, 24, 2007, pp. 112–122.
[15] S. Mukherjee, Architecture design for soft errors, Morgan Kauf (2008).
[16] A. Ptzmann, M. Hansen, Anonymity unlinkability undetectability unobservability pseudonymity and identity management, A Consolidated Proposal for Terminology version 0.31, 2008.
[17] R. Ubar, et al., Diagnostic modeling of digital systems with multi-level DDs, in: R. Ubar, J. Raik, Vierhaus H.Th. (Eds.), Design and Test Technology For Dependable, SoC, 2011, pp. 92–118.
[18] V. Guarnieri, Mutation analysis for SystemC designs at TLM, in: 2011 12th Latin American Test Workshop (LATW), Porto de Galinhas, 2011, pp. 1–6.
[19] J. Hansson, B. Lewis, J. Hugues, L. Wrage, P. Feiler, J. Morley, Model-Based verification of security and non-functional behavior using AADL, IEEE Secur. Priv. (2009) 1–1.
[20] J. Sepulveda, D. Aboul-Hassan, G. Sigl, B. Becker, M. Sauer, Towards the formal verification of security properties of a Network-on-Chip router, in: 2018 IEEE 23rd European Test Symposium (ETS), Bremen, 2018, pp. 1–6, doi:10.1109/ETS.2018.8400692.
[21] T. Boraten, D. DiTomaso, A.K. Kodi, Secure model checkers for Network-on-Chip (NoC) architectures, in: 2016 Int. Great Lakes Symposium on VLSI (GLSVLSI), Boston, MA, 2016, pp. 45–50.
[22] H.M.G. Wassel, Networks on chip with provable security properties, IEEE Micro. 34 (3) (May-June 2014) 57–68.
[23] M.A. Kochte, M. Sauer, L.R. Gomez, P. Raiola, B. Becker, H.J. Wunderlich, Specification and verification of security in reconfigurable scan networks, in: 2017 22nd IEEE European Test Symposium (ETS), Limassol, 2017, pp. 1–6.
[24] L.W. Kim, J.D. Villasenor, Dynamic function verification for system on chip security against hardware-based attacks, IEEE Transa. Reliab. 64 (4) (Dec. 2015) 1229–1242.
[25] Hu Wei, et al., Imprecise security: quality and complexity tradeoffs for hardware information flow tracking, in: IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), Austin, TX, 2016, pp. 1–8.
[26] A. Nahiyan, M. Sadi, R. Vittal, G. Contreras, D. Forte, M. Tehranipoor, Hardware trojan detection through information flow security verification, in: 2017 IEEE International Test Conference (ITC), Fort Worth, TX, 2017, pp. 1–10.
[27] M.A. Kochte, R. Baranowski, M. Sauer, B. Becker, H.J. Wunderlich, Formal verification of secure reconfigurable scan network infrastructure, in: 2016 21th IEEE European Test Symposium (ETS), Amsterdam, 2016, pp. 1–6.
[28] M. Rocchetto, N.O. Tippenhauer, Towards formal security analysis of industrial control systems, in: ACMA sia Conf. Comput. Commun. Secur., 2017, pp. 114–126.
[29] M. Yoshimura, T. Bouyashiki, T. Hosokawa, A hardware trojan circuit detection method using activation sequence generations, in: 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), Christchurch, 2017, pp. 221–222.
[30] F.K. Lodhi, S.R. Hasan, O. Hasan, F. Awwad, Formal analysis of macro synchronous micro asychronous pipeline for hardware trojan detection, in: NORCAS, Oslo, 2015, pp. 1–4.
[31] S. Deng, W. Xiong, J. Szefer, Cache timing side-channel vulnerability checking with computation tree logic, in: Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy, ser. HASP '18, New York, NY, USA, ACM, 2018, p. 2. 1–2:8.
[32] T. Zhang, R.B. Lee, New models of cache architectures characterizing information leakage from cache side channels, in: Proceedings of the 30th Annual Computer Security Applications Conference, ser. ACSAC'14, New York, NY, USA, ACM, 2014, pp. 96–105.
[33] P. Canones, B. Köpf, J. Reineke, Security analysis of cache replacement policies, CoRR, 2017 arXiv:1701.06481.

[34] Z. Hanna, Verifying security aspects of SoC designs with Jasper app, (white paper), Jasper Design Automation (Cadence), 2013.
[35] JasperGold Security Path Verification App, Cadence, http://www.cadence.com.
[36] J. Vain, A. Kaur, L. Tsiopoulos, J. Raik, M. Jenihhin, Multi-view modeling for MP-SoC design aspects, in: 2018 16th Biennial Baltic Electronics Conference (BEC), Tallinn, 2018, pp. 1–6.
[37] Q. Ge, Y. Yarom, D. Cock, G. Heiser, A survey of microarchitectural timing attacks and countermeasures on contemporary hardware, J. Cryptogr. Eng. 8 (1) (2018) 1–27.
[38] Y. Lyu, P. Mishra, A survey of side-channel attacks on caches and countermeasures, J. Hardw. Syst. Secur. 2 (1) (Mar 2018) 33–50.
[39] A. Savino, S. Di Carlo, A. Vallero, G. Politano, D. Gizopoulos, A. Evans, RIIF-2: toward the next generation reliability information interchange format, IEEE IOLTS (2016) 173–178.
[40] C. Liu, J. Layland, Scheduling algorithms for multi programming in a hard real time environment, J. ACM 20 (1973) 46–61.
[41] F.S. Gonçalves, D. Pereira, E. Tovar, L.B. Becker, Formal verification of AADL models using UPPAAL, in: 2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC), Curitiba, 2017, pp. 117–124.
[42] D. Burlyaev, P. Fradet, Formal verification of automatic circuit transformations for fault-tolerance, in: 2015 Formal Methods in Computer-Aided Design (FMCAD), Austin, TX, 2015, pp. 41–48.
[43] B.W. Thompto, B. Hoppe, Verification for fault tolerance of the IBM system z microprocessor, in: Design Automation Conference, Anaheim, CA, 2010, pp. 525–530.
[44] S. Kan, M. Lam, T. Porter, J. Dworak, A case Study: pre-Silicon SoC RAS validation for NoC server processor, in: MTV, 2016, pp. 19–24.
[45] S. Vinco, M. Lora, E. Macii, M. Poncino, IP-XACT for smart systems design: extensions for the integration of functional and extra-functional models, in: 2016 Forum on Specification and Design Languages (FDL), Bremen, 2016, pp. 1–8.
[46] S. Vinco, Y. Chen, F. Fummi, E. Macii, M. Poncino, A layered methodology for the simulation of extra-functional properties in smart systems, in: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36, 2017, pp. 1702–1715.
[47] G. Aleksandrowicz, et al., Designing reliable cyber-physical systems, in: F. Fummi, R. Wille (Eds.), Languages, Design Methods, and Tools for Electronic System Design. Lecture Notes in Electrical Engineering, 454, Springer, Cham, 2018.
[48] Eli Arbel, Shlomit Koyfman, Prabhakar Kudva, Shiri Moran, Automated detection and verification of parity-protected memory elements, in: Proc. IEEE/ACM ICCAD, 2014, pp. 1–8.
[49] M. Maniatakos, Y. Makris, Workload-driven selective hardening of control state elements in modern microprocessors, in: VTS, 2010, pp. 159–164.
[50] S. Ganapathy, R. Canal, D. Alexandrescu, E. Costenaro, A. González, A. Rubio, INFORMER: an integrated framework for early-stage memory robustness analysis, in: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014, pp. 1–4.
[51] S. Avramenko, S.P. Azad, S. Esposito, B. Niazmand, M. Violante, J. Raik, M. Jenihhin, QoSinNoC: analysis of qos-Aware NoC architectures for mixed-criticality applications, in: 21st IEEE Int. Symp. DDECS, 2018, pp. 1–6.
[52] S. Avramenko, Upgrading QoSinNoC: efficient routing for mixed-criticality applications and power analysis, in: IEEE VLSI-SoC, Verona, 2018, pp. 1–6.
[53] S. Rubini, F. Singhoff, J. Hugues, Modeling and verification of memory architectures with AADL and REAL, in: 2011 16th IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, NV, 2011, pp. 338–343.
[54] H. Wang, X. Zhou, Y. Dong, L. Tang, A hierarchical verification procedure of timed petri-net model for real-time embedded systems, in: 2010 2nd International Conference on Information Engineering and Computer Science, Wuhan, 2010, pp. 1–4.
[55] H. Wang, X. Zhou, Y. Dong, L. Tang, Timing properties analysis of real-time embedded systems with AADL model using model check, in: IEEE Int. Conf. on Progress in Informatics and Computing (PIC), 2010, pp. 1019–1023.
[56] A. Rafiev, F. Xia, A. Iliasov, A. Romanovsky, A. Yakovlev, Selective abstraction for estimating extra-functional properties in Networks-on-Chips using archon framework, in: 2017 17th International Conference on Application of Concurrency to System Design (ACSD), Zaragoza, 2017, pp. 80–85.
[57] P. Khondkar, Low-Power Design and Power-Aware Verification, Springer, 2018.
[58] D. Lorenz, Non-invasive power simulation at system-level with systemc, PATMOS 2012. LNCS (7606), Springer, 2012.
[59] S. Orcioni, et al., Energy estimation in SystemC with powersim, Integr. VLSI J. (55) (2016) 118–128.
[60] ANSI/IEEE 1801-2015 - IEEE Standard for design and verification of Low-Power, energy-aware electronic systems, March 2016,
[61] Si2 Common Power Format, v2.1, Silicon Integration Initiative, 2014.
[62] E.Y. Kang, D. Mu, L. Huang, Q. Lan, Verification and validation of a cyber-physical system in the automotive domain, in: 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, 2017, pp. 326–333.
[63] A. Danese, G. Pravadelli, I. Zandonà, Automatic generation of power state machines through dynamic mining of temporal assertions, in: 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 606–611.
[64] J. Zimmermann, S. Stattelmann, A. Viehl, O. Bringmann, W. Rosenstiel, Model-driven virtual prototyping for real-time simulation of distributed embedded systems, in: 7th IEEE Int. Symposium on Industrial Embedded Systems (SIES'12), Karlsruhe, 2012, pp. 201–210.
[65] R. Görgen, et al., CONTREX: design of embedded mixed-criticality CONTRol systems under consideration of EXtra-Functional properties, in: 2016 Euromicro Conference on Digital System Design (DSD), Limassol, 2016, pp. 286–293.
[66] A.W. Ruan, Y.B. Liao, P. Li, W.C. Li, W. Li, Throughput estimation for modelsim simulator tool based HW/SW co-verification system, in: 2009 International Conference on Communications, Circuits and Systems, Milpitas, CA, 2009, pp. 1014–1018.
[67] M. Khamis, S. El-Ashry, A. Shalaby, M. AbdElsalam, M.W. El-Kharashi, A configurable RISC-V for noc-Based MPSoCs: a framework for hardware emulation, in: 2018 11th International Workshop on Network on Chip Architectures (NoCArc), Fukuoka, 2018, pp. 1–6.
[68] JasperGold Connectivity Verification App, Cadence, http://www.cadence.com.
[69] S.K. Roy, Top level SOC interconnectivity verification using formal techniques, in: The 8th Int. Workshop on Microprocessor Test and Verification, Austin, TX, USA, 2008, pp. 63–70.
[70] M. Elver, V. Nagarajan, McVerSi: a test generation framework for fast memory consistency verification in simulation, in: 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), Barcelona, 2016, pp. 618–630.
[71] S.-P. Azad, B. Niazmand, K. Janson, J. Raik, Github Bonfire Project (2017), https://github.com/Project-Bonfire/ (accessed 1 June 2019).
[72] S.P. Azad, From online fault detection to fault management in Network-on-Chips: a ground-up approach, in: 2017 IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Dresden, 2017, pp. 48–53.
[73] S.-C. Fang, C.-C. Weng, C.-K. Tseng, C.-W. Hsu, J.-L. Liao, S.-Y. Huang, C.-L. Lung, D.-M. Kwai, SoC power analysis framework and its application to power-thermal co-simulation, in: 2011 Int. Symp. on VLSI Design, Automation and Test, April 2011, pp. 1–4.
[74] G. Vece, M. Conti, S. Orcioni, Transaction-level power analysis of VLSI digital systems, Integr. VLSI J. 50 (2015) 116–126.
[75] M. Giammarini, M. Conti, S. Orcioni, System-level energy estimation with powersim, in: 2011 18th IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS), December 2011, pp. 723–726.
[76] Bell Labs, Verifying Multi-threaded Software with Spin, (1980). http://spinroot.com/ (accessed 1 June 2019).
[77] SMT Steering Committee, The International Satisfiability Modulo Theories (SMT) Competition. http://www.smtcomp.org/ (accessed 1 June 2019).
[78] Carnegie Mellon University, Architecture Analysis and Design Language. http://www.aadl.info/aadllcurrentsite/ (accessed 1 June 2019).
[79] S. Dai, Y. Zhou, H. Zhang, E. Ustun, E.F.Y. Young, Z. Zhang, Fast and accurate estimation of quality of results in high-level synthesis with machine learning, in: 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2018, pp. 129–132.
[80] K. Hasegawa, M. Oya, M. Yanagisawa, N. Togawa, Hardware Trojans classification for gate-level netlists based on machine learning, in: 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), 2016, pp. 203–206.
[81] K. Hasegawa, M. Yanagisawa, N. Togawa, Hardware Trojans classification for gate-level netlists using multi-layer neural networks, in: 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), 2017, pp. 227–232.
[82] E. Alpaydin, F. Bach, Introduction to Machine Learning, MIT Press, 2014.
[83] B. Yu, D.Z. Pan, T. Matsunawa, X. Zeng, Machine learning and pattern matching in physical design, in: The 20th Asia and South Pacific Design Automation Conference, 2015, pp. 286–293.
[84] B. Li, P.D. Franzon, Machine learning in physical design, in: 2016 IEEE 25th Conference on Electrical Performance Of Electronic Packaging And Systems (EPEPS), 2016, pp. 147–150.
[85] L. Bai, L. Chen, Machine-Learning-Based early-stage timing prediction in SoC physical design, in: 2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2018, pp. 1–3.
[86] K. Hasegawa, M. Yanagisawa, N. Togawa, Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier, in: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017, pp. 1–4.

Xinhui Lai is one of the early stages researchers in the RESCUE European Training Network. She is doing her research work and Ph.D in Tallinn University of Technology which is one of the institutions evolved in the RESCUE project. She got her bachelor and master diploma in Electronic Engineering from Politecnico di Torino, Italy, in October 2014 and April 2017 respectively. Her research is focused on design error functional verification and automated debug, i.e. localization and correction, as well as verification of extra-functional interdependent aspects in nanoelectronic system design such as security, reliability, power/performance envelope.

Aneesh Balakrishnan is an Early Stage Researcher in the RESCUE European Training Network. Within the project, he is a Research and Development Engineer at iRoC Technologies, France and also a Ph.D. Candidate in the department of computer systems at Tallinn University of Technology, Estonia. Aneesh has a master degree in Communication and Multimedia Engineering from Friedrich-Alexander University, Erlangen-Nurnberg, Germany in July 2016 and holds a Bachelor of Engineering in Electronics and Communication Engineering from Anna University of Technology, India. His current research will address today's high-performance designs requirements in term of validation and reliability. The objective of the research is to significantly enhance and develop new statistical, probabilistic methods and algorithms.

Thomas Lange is an Early Stage Researcher in the RESCUE European Training Network. Within the project he is a Research and Development Engineer at iRoC Technologies, France and also Ph.D. Candidate in Computer and Systems Engineering at Politecnico di Torino, Italy. Thomas holds a Bachelor's and Master's degree in Computer Engineering from Technische Universität Berlin. In his research he is investigating the effects of transient faults for high reliability applications in harsh environments. His main interest includes the development of new models and assessment techniques for transient faults, as well as new mitigation and error management techniques with the focus on hardware capabilities.

Maksim Jenihhin is a professor of Computing Systems Reliability at the Department of Computer Systems of Tallinn UT. He holds M.*Sc.* and Ph.D. degrees in Computer Engineering from Tallinn UT (2004 and 2008 respectively). His research interests include methodologies and EDA tools for hardware design, verification and debug as well as nanoelectronics reliability and manufacturing test topics. Maksim is a project coordinator for H2020 RESCUE - Interdependent Challenges of Reliability, Security and Quality in Nanoelectronic Systems Design.

Tara Ghasempouri is a Postdoctoral Researcher at Tallinn University of Technology in Computer Systems department. Her group is mainly focused on three categories such as fault tolerance, verification and safety/security of systems. She is interested in finding innovative solutions for the verification process at the different level of abstraction. Her research topic is also focused on Hardware Security. She received a Ph.D. degree in Computer Science from University of Verona, Italy. During her Ph.D. program, she has researched on different kind of verifications and specially Assertion-based Verification on the embedded system. She is a member of IEEE Computer Society and she was a reviewer for different conferences such as ETS, VLSI-SOC and etc.

Jaan Raik is a professor of digital systems verification at the Department of Computer Systems of Tallinn University of Technology and the leader of the Center for Dependable Computing Systems Design (DCSD). Prof. Raik received his M.*Sc.* and Ph.D. degrees in Computer Engineering from Tallinn University of Technology in 1997 and in 2001, respectively. He is a member of IEEE Computer Society, HiPEAC and a member of steering/program committees of several conferences. He has co-authored more than 200 scientific publications.

Dan Alexandrescu (IEEE Member'07-Senior Member'13) is the CEO of IROC Technologies. Dan holds a M. Eng. in Electronics from Politehnica Bucharest, Romania, a M.A.S. in Microelectronics from Joseph Fourier University, Grenoble, France and a Ph.D in Microelectronics from INPG, Grenoble Institute of Technology, France. He specializes in the design, optimization and improvement of highly-reliable microelectronic circuits. He contributed to the organization of reliability-centric workshops and symposia (Program Co-Chair for multiple IOLTS editions, Finance and General Co-Chair for several SELSE editions) and he prepared many publications in the field of reliability and radiation-induced effects.