Early Functional Safety and PPA Evaluation for Faster Digital Design Development

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 April 2026, accepted 28 April 2026, date of publication 4 May 2026, date of current version 11 May 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3690084

RESEARCH ARTICLE

Early Functional Safety and PPA Evaluation for Faster Digital Design Development

MICHELANGELO BARTOLOMUCCI[1], (Graduate Student Member, IEEE), DAVID KINGSTON[2], TEO CUPAIUOLO[2], ALESSANDRA NARDI[2], AND RICCARDO CANTORO[1], (Member, IEEE)

[1]Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy
[2]Synopsys, Inc., Sunnyvale, CA 94085, USA

Corresponding author: Michelangelo Bartolomucci (michelangelo.bartolomucci@polito.it)

This work was supported in part by the Project Piano Nazionale di Ripresa e Resilienza – NextGenerationEU (PNRR-NGEU) from Ministero dell'Università e della Ricerca (MUR) under Grant DM 117/2023.

ABSTRACT The use of semiconductor devices in safety-critical applications is increasing in both volume and complexity. Applications in markets such as automotive, data centers, and aerospace have dependability requirements that impose additional safety and reliability metrics beyond the traditional Power, Performance, and Area (PPA). Semiconductor devices for safety-critical applications incorporate protection logic to enhance resilience against random failures (i.e., they are hardened). Although automation to optimize PPA and Time-to-Market (TTM) is well established, further effort is required to incorporate safety as a first-class metric in the design process. This paper presents a novel approach that supports safety requirements from Register Transfer Level (RTL) exploration through implementation, thereby minimizing costly design iterations. A Safety Specification Format (SSF) is used to explore different protection levels and safety mechanisms at the RTL stage and to seamlessly implement the selected configuration. The approach is evaluated on the CV32E40P and CVA6 open-source RISC-V processors and on ISCAS89 benchmarks. Validation is performed through trend comparison and quantitative correlation against traditional implementation methodologies across multiple designs.

INDEX TERMS Digital design process, functional safety, shift-left, RTL exploration, safety-critical systems, selective hardening, safety specification format, PPA, SSF.

I. INTRODUCTION

Very Large Scale Integration (VLSI) circuits are growing in complexity, now reaching transistor counts in the trillions [1]. Developing these circuits requires multiple iterations to achieve the target Power, Performance, and Area (PPA) metrics detailed in the specifications. To report PPA data, the circuit must undergo the implementation process, specifically through logic and physical synthesis at the very least. Modern designs, such as System-on-Chips (SoCs), require significant synthesis time, and performing this process iteratively can significantly extend the Time-to-Market (TTM) for the product.

Circuit hardening is often also required to comply with Functional Safety (FuSa) standards [2], [3]. Achieving a target Single Point Fault Metric (SPFM) coverage, especially for very critical designs (e.g., 99% SPFM coverage), can significantly impact the synthesis and, by extension, the overall development time of the system. PPA metrics often degrade when hardening the circuit compared to the unhardened version. This degradation requires additional effort and runtime by the tools to contain the PPA degradation within the specified limits.

Electronic Design Automation (EDA) vendors have proposed solutions to address the problem of extensive runtime. In particular, some post-implementation metrics estimators have been developed [4]. These solutions aim to predict the final PPA and congestion metrics of the implementation within a certain correlation range. These technologies use faster, predictive synthesis techniques compared to actual synthesis optimization tools, enabling the completion of design mapping, optimization, and floorplanning exploration in significantly reduced runtimes. To avoid discrepancies, the estimators can process the scripts used for actual synthesis. Estimators can also predict the impact of a given hardening in a circuit, making them useful in contexts where FuSa techniques must be applied.

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

As digital designers often need to make several design choices at the high-level code stage, exploiting a faster approach for the exploration phase drastically reduces overall development time. Implementation of FuSa requirements is traditionally performed after the synthesis phase. Hardening a circuit can thus extend the development time, since complying with PPA constraints in late stages of development becomes more challenging. Being able to prototype the insertion of Safety Mechanisms (SMs) and assess their relative PPA impact in the early development stages can enable digital designers to produce hardening-oriented circuits and reduce the effort required in late development stages, directly reducing development iterations and, by extension, TTM.

Additionally, EDA-oriented predictive methodologies based on Artificial Intelligence (AI), and unrelated to FuSa, have been recently introduced in literature [5], [6], [7], [8], [9]. However, due to the probabilistic nature of the technology, introducing AI-based methodologies in FuSa flows can introduce some non-determinism, which is not regulated by current standards.

This paper introduces an enhanced safety-aware digital design implementation flow, enabling early design exploration of PPA together with safety and reliability metrics. The goal is to quickly assess the safety metrics and their impact on PPA to define the final strategy for achieving the required Automotive Safety Integrity Level (ASIL), while consistently reducing runtimes in the iterative process. The SMs to be explored are captured using a Safety Specification Format (SSF). Once the optimal configuration is decided, the flow continues to implementation, based on the selected Register Transfer Level (RTL) plus SSF configuration. In addition, for transient errors (i.e., Soft Errors), the flow can start from an analysis step that identifies critical registers to be protected to achieve the desired metrics.

We implemented the proposed methodology on open-source RISC-V cores and sequential benchmark circuits. We evaluated multiple scenarios targeting different safety mechanisms and various ASIL levels, comparing the PPA of the unhardened and the hardened designs. Validation is performed through implementation. Commercial EDA tools have been employed in the flow's methodology, but others (e.g., open-source) can be easily integrated.

The rest of the paper is organized as follows: Section II provides context and related works. Section III summarizes the motivation and purpose of the proposed methodology. The proposed approach is presented in Section IV. Section V and Section VI describe the experimental setup and discuss the obtained results. Finally, Section VII offers some conclusions.

II. BACKGROUND

A. FUNCTIONAL SAFETY

Functional Safety (FuSa) is formally defined in the IEC 61508 standard [2]. This generic standard serves as the basis for industry-specific standards, such as ISO 26262 for the automotive industry. ISO 26262 defines FuSa as the "absence of unreasonable risk due to hazards caused by malfunctioning behavior of Electrical/Electronic systems" [3].

Various metrics are defined in the standard to characterize systems [10]. The SPFM measures the resilience of a component to single-point faults, while the Latent Fault Metric (LFM) measures resilience to latent faults. These metrics can be evaluated based on the design implementation and the fault coverage of the implemented mechanisms and are expressed as percentages. Additionally, the Probabilistic Metric of Hardware Failures (PMHF) reports the system's robustness to random hardware faults, expressed in Failures in Time (FIT), which is the number of failures expected in one billion hours of operation.

Compliance thresholds for these metrics, required for system certification, are defined through the ASIL specification. ASIL-A represents the lowest level of risk reduction, while ASIL-D represents the highest. Table 1 shows the different compliance thresholds for the three metrics.

TABLE 1. ISO 26262 failure metrics specification.

ASIL	PMHF [FIT]	SPFM [%]	LFM [%]
A	< 1000	-	-
B	< 100	≥ 90	≥ 60
C	< 100	≥ 97	≥ 80
D	< 10	≥ 99	≥ 90

Different analyses are performed on systems to ensure compliance. The most relevant analyses include Failure Modes Effect and Diagnosis Analysis (FMEDA), Timing Analysis, and Dependent Failure Analysis (DFA). FMEDA defines the components' Failure Modes (FM) and diagnostic capabilities. It divides the component into various subparts, specifies the respective FMs for each subpart, and enumerates the SMs implemented to address them. Finally, a Diagnostic Coverage value is provided, indicating the percentage of FM-related faults that can be identified by the SMs.

In [11], the authors describe the FuSa Analysis Flow. The first step is a Qualitative Analysis based on the functional and block-level description of the circuit, primarily aimed at identifying potential failure modes. Once the circuit is implemented, a Quantitative (FMEDA) Analysis is performed. After linking the FMs identified in the previous stage to the actual components, their weights and failure probabilities are used as a basis for predicting the Failure Mode Distribution.

Functional Safety targets can be achieved using various techniques. SMs are employed to ensure online reliability. Hardware, information, and time redundancy are possible methodologies to detect, mask, and reduce the error risk. Diagnostic mechanisms can also be included [12]. Hardware strategies include Triple Modular Redundancy (TMR), suitable for protecting pipelines and small groups of registers, and Error Correction Codes (ECC), better suited for larger memory structures (e.g., Register Files and SRAMs) [13]. Memory cells that are intrinsically resilient to soft errors,

such as the Dual Interlocked Storage Cell (DICE), have been proposed in the literature [14]. However, for deep sub-micron technology nodes, layout-level adaptations are required [15].

B. SELECTIVE HARDENING

Implementing SMs can be costly in terms of PPA metrics, especially when applied to the entire design. Methodologies for selective hardening (i.e., targeted SM insertion in only the critical parts of the design) have been proposed in the literature. In [16], the authors provide an overview of the problem, discussing general and special (e.g., NanoPLA) circuits.

Safety analysis is essential for implementing selective circuit hardening as it identifies which modules to protect with SMs based on their impact on FuSa metrics. In particular, for the SPFM, an important contributing factor is the Soft Error Rate (SER) of each component. An SER analysis of the circuit is therefore necessary to estimate the safety impact of the different components. In the literature, multiple works have been proposed [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30] including both dynamic and static analysis approaches.

Dynamic analysis requires large sets of input vector data to drive simulations, which may not be fully available early in the design process or even known at all for highly configurable components. Although each individual simulation is accurate, the overall results are sensitive to when a fault is injected. Moreover, confidence in transient fault injection results is inherently tied to the confidence in functional verification coverage. Runtime is also a significant concern, further exacerbated by the dependency on injection timing. In contrast, static analysis does not require input vectors or simulation. While it can be less accurate, it avoids the coverage and runtime limitations of dynamic approaches. In [30], a statistical method is introduced that leverages the strengths of static analysis while expanding it to "exhaustively" cover the workload space, thereby improving accuracy and providing deeper insight into design optimization tradeoffs. When both static and dynamic techniques are available, they can be combined to achieve the best results depending on the overall methodology. However, because this paper focuses on early stage design exploration, the characteristics of static analysis make it the more suitable choice for the proposed approach.

Additionally, several approaches focusing on selective hardening have been proposed. In [31], a solution for early design stages, when accurate timing and electrical information are not yet available, is proposed. This approach leverages Binary Decision Diagrams (BDDs) to compute the probability of error in the components. In [32], a scalable but approximate selective hardening approach is proposed that focuses on computing the logical masking probability of a given node. Additionally, [33] proposes a method based on forward and backward traversing techniques to automatically identify the components contributing most to Failure Rates. However, this method relies on user-generated input vectors, which can hinder a fully automated flow. Unlike these approaches, which operate at the Gate-Level and require logic synthesis, our approach focuses on the RTL stage, avoiding technology-specific analyses and reducing the overall runtime of FuSa analyses and implementation.

In [34], the authors propose a fully automated flow for selective hardening using TMR as an SM. This graph-based approach generates two sub-graphs related to unhardened and hardened registers, focusing on achieving the target reliability level with the smallest possible TMR-related subgraph. Although promising results in terms of area reduction are reported, the iterative partitioning process involves multiple simulations, significantly increasing the overall runtime.

Finally, in [35], an interesting application of selective hardening is proposed for GPUs to ensure correct instruction execution. This approach involves fault injection campaigns on the RTL description of the GPU, classifying faults based on their probability of reaching the outputs, and processing them through a feedback unit. This unit can trace errors back to the source code, identifying candidates for selective hardening. While promising, this iterative process involves multiple simulations, potentially increasing overall runtime.

C. SAFETY SPECIFICATION FORMAT

The Safety Specification Format (SSF) [36] is a set of commands designed to facilitate the implementation of FuSa requirements across all phases of implementation and verification and across different tools. The SSF captures design intent across all the stages of the implementation process, starting from the RTL design down to the physical implementation. This provides designers with an end-to-end strategy for tracking FuSa requirements and their implementation. The SSF addresses different stages of the FuSa process. Common SMs (e.g., TMR, ECC, Dual-Core Lockstep) are natively supported in the language. For corner cases, it is also possible to define custom SMs comprising hardened cells. The intent for SM insertion during the flow can also be specified. Applying a single SM to the entire circuit, performing selective hardening on specific parts, or applying different SMs to different sub-modules can all be easily accomplished through intent definitions. Additionally, it is possible to identify and mark SMs that were pre-inserted using previous or alternative flows. In this paper, the SSF methodology is used to abstract low-level FuSa implementation details, which are not relevant to our research, and to facilitate SM prototyping at the RTL stage, which would otherwise require manual effort. Using SSF provides an automated way to insert SMs into circuits, ensuring consistency between the RTL and final implementation stages, thereby further validating the proposed methodology.

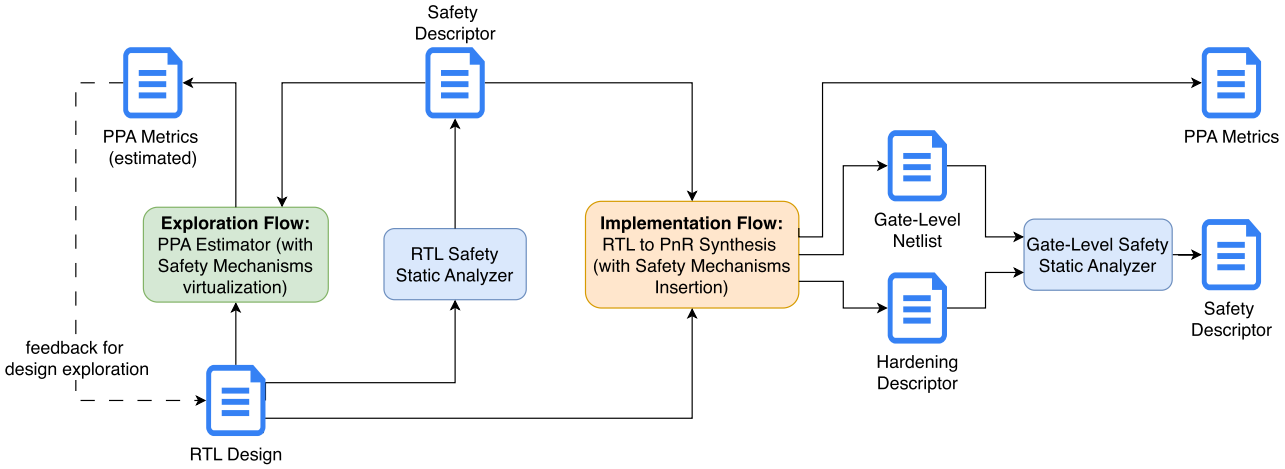


FIGURE 1. Proposed flow for design exploration and implementation.

III. MOTIVATION

Performing synthesis on modern, complex designs is a time-consuming process. This limits the possible design explorations that an RTL designer can perform without heavily impacting the TTM.

Additionally, hardening a design for FuSa (i.e., applying SMs) to meet safety metric targets can have a significant impact on PPA. As a result, often only selected critical components are hardened, and trade-offs are made regarding the type of SMs: different SM types, such as Triple Modular Redundancy (TMR) or Parity-based schemes, affect PPA differently (e.g., TMR typically incurs greater area overhead than Parity-based SMs).

However, traditional methods for identifying safety-critical registers, such as fault injection, require significant runtime and depend on mature verification environments being in place. This often forces designers to choose between lengthy simulations late in the development cycle and relying on expert judgment.

SM insertion is typically performed either post-synthesis at the Gate-Level, which bypasses crucial PPA optimizations, or at the RTL, which locks the protection strategy early, before knowledge of the specific workload may be available. Both approaches limit flexibility in exploring trade-offs between safety metrics and PPA.

Furthermore, modern CPU and SoC designs are highly modular and configurable. At the RTL stage, understanding the impact of such modifications on the final design is often difficult, and estimates are prone to significant error. Performing early analysis helps designers anticipate the impact of enabling specific features or modules on final PPA, thereby helping ensure the design remains within specified constraints.

Clearly, designers face significant challenges in meeting safety goals while achieving PPA closure. The lack of infrastructure for early identification of critical registers and systematic exploration of SM options exacerbates these challenges.

IV. PROPOSED APPROACH

To address these challenges, we propose a multistage automated flow, shown both graphically, in Figure 1, and as pseudocode, in Algorithm 1.

This flow aims to guide front-end RTL designers in understanding the impact of hardening on both safety and PPA, selecting the optimal architecture upfront, and then automating the implementation.

A. SAFETY STATIC ANALYSIS

The first stage performs a static safety analysis to compute the controllability and observability of circuit components, particularly registers, and to quantify each component's contribution to SPFM loss. These analyses operate directly on the RTL description and do not require a verification environment or logic synthesis, making them independent of the implementation library and technology node. Several analysis approaches exist. However, static analysis is uniquely well suited for early stage estimation, where input vectors, verification coverage, and implementation details may not be available yet. The proposed methodology remains agnostic to how SPFM loss is generated: dynamic simulation based data can be used when available, and RTL simulation data can further augment the static results. The SPFM data produced by the tool, exported as an SSF file, serves as the foundation for circuit hardening decisions in subsequent stages. The SSF exchange file format uses a set of Tcl commands to describe the safety-related aspects of a design for use with safety-aware implementation and verification flows. The information conveyed through SSF includes design properties (e.g., SPFM loss contribution for a given cell) and intents (e.g., applying an SM to a cell or marking a cell as part of an SM).

B. EXPLORATION

In the exploration phase, the SSF is provided to an RTL implementation metrics estimator. Because this tool is built on the same engine as the RTL to Place-and-Route (PnR)

synthesis tool, its results are consistent with implementation, and PPA trends are preserved. Unlike standard RTL parsers, which provide only rough estimates of standard cell counts, the RTL implementation metrics estimator can evaluate a complete set of metrics up to the PnR stage using fast synthesis techniques. This fast synthesis flow begins with a mapping and optimization stage, during which cells are mapped and area and timing optimizations are performed. It then proceeds to automatic floorplanning, during which the core area is defined based on the mapped netlist, and macro and port placement is performed. Next, placement and physical-aware synthesis are performed, including clock tree buffering and auto-routing. The process concludes with the computation of final metrics using timing and power static analyzers. The exploration identifies the optimal PPA and safety metrics scenario, and the chosen architecture is captured as an RTL plus SSF configuration.

C. IMPLEMENTATION

Once the exploration phase is complete, the selected RTL plus SSF configuration is passed to the implementation flow to generate the final post-PnR netlist. Since the exploration and implementation tools share the same engine, the scripts are largely interchangeable, eliminating potential inconsistencies between the design configuration and elaboration stages. In addition to PPA optimization, the SMs defined in the SSF are inserted during synthesis and correctly implemented during PnR, adhering to safety requirements to avoid common-cause failures.

D. FINAL SAFETY CHECK

The implementation phase can be followed by a final safety check using the safety static analyzer in Gate-Level mode to verify the SPFM coverage achieved by the resulting netlist. The safety analyzer uses the Hardening Descriptor file (an SSF file describing all SMs inserted during the exploration and implementation phases) to identify which components are part of the hardening and to compute the final SPFM coverage.

E. VALIDATION

The proposed methodology is validated through trend comparison and quantitative correlation between the two tools, demonstrating the consistency between estimated metrics and implementation results.

Unlike the industry-standard approach, the proposed flow emphasizes performing development iterations during the exploration phase rather than during implementation, thereby reducing TTM.

V. EXPERIMENTAL SETUP

The flow has been implemented using a Synopsys synthesis and implementation methodology. The proposed flow, however, is tool-agnostic and can be implemented with other EDA tools. Additionally, the SSF format has been employed to capture the safety intent to prototype and implement the safety mechanisms, ensuring coherence of usage and refinement of information across design stages. However, in the case of open-source alternatives for tools, also the SSF format can be replaced with other ways of capturing the same information. Note that keeping the safety intent separate from the RTL has the advantage of exploration and creating different safety strategies for different ASILs.

For implementation metrics estimation, we used Synopsys *RTL Architect* (RTLA), version X-2025.06-SP1. The tool performs a unified flow (i.e., from logic synthesis and optimization through floorplanning and physical implementation). It requires a unified library built using the New Data Model (NDM). This library format combines all physical and logical information about the library's components into a single binary, whereas traditional formats distribute this information across multiple files, such as Library Exchange Format (LEF), Liberty Timing (.lib), and Technology files.

To identify the critical parts of the circuit, we employed Synopsys *VC SpyGlass Fault Analysis*, version X-2025.06-SP1-1, as the RTL and Gate-Level static safety analyzer. By default, this tool checks only the observability of registers through the top-level entity's output ports. This default analysis is limited and often results in unrealistically high SPFM coverage for unhardened circuits. To provide more realistic analysis and create more challenging scenarios, additional internal observation points were added to the tool's scope.

For RTL to PnR implementation, we used Synopsys *Fusion Compiler*, version X-2025.06-SP1, with the same NDM library and synthesis scripts as RTLA.

Experiments were conducted on several circuits. The *CV32E40P* and *CVA6* from OpenHW Group [37], [38] were selected as open-source RISC-V cores. Additionally, experiments were performed on selected ISCAS89 benchmarks [39]. The CV32E40P is a 4-stage, in-order, single-issue pipelined processor with a 32-bit data bus. The CVA6 is a 6-stage, fully pipelined, in-order, single-issue processor with a 64-bit data bus. The selected ISCAS89 benchmarks are listed in Table 2. The Synopsys SAED32 library [40], developed for the 32nm technology node and available with multiple process corners, was used for synthesis. For our experiments, we used the slow process corner with a supply voltage of 0.95V and a temperature of −40°C.

TABLE 2. Selected circuits for methodology exploration and validation.

Circuit	#DFFs	#STD_CELLs	Clock Period [ns]
s9234	138	771	0.50
s13207	223	1,056	0.25
s38584	893	4,175	0.50
s38417	1,421	5,746	0.25
s35932	1,502	5,711	0.25
cv32e40p	2,209	14,779	1.00
cva6	23,793	163,484	2.50

For the exploration experiments, multiple targets were defined to produce netlists with varying safety constraints. Multiple safety thresholds were targeted, ranging from

Algorithm 1 Proposed Shift-Left Methodology Pseudocode

Require: File list, PPA constraint file
Ensure: Implementation netlist, Safety descriptor

— Initial analyses —

design ← READDESIGN(*filelist*)
RUNRTLSAFETYANALYSIS(*design*, *spfm_data.ssf*)
ppa_constraints ← READPPACONSTRAINTS(*ppa_constraints_file*)

— Early Design Exploration —

exploration_metrics ← RUNExploration(*design*, *spfm_data.ssf*)
while ¬CHECKCONSTRAINTS(*ppa_constraints*, *exploration_metrics*) **do**
    design ← REVISEDESIGN(*design*)
    exploration_metrics ← RUNEXPLORATION(*design*, *spfm_data.ssf*)
end while

— Final Implementation —

(*implementation_metrics*, *netlist*) ←
    RUNIMPLEMENTATION(*design*, *spfm_data.ssf*, *hardening_descriptor.ssf*)
RUNGATELEVELSAFETYANALYSIS(*netlist*, *hardening_descriptor.ssf*, *safety_descriptor.ssf*)

ASIL-B compliance (90% SPFM coverage) to ASIL-D compliance (99% SPFM coverage), as well as an unhardened baseline. For each safety threshold, multiple SM types were implemented, including TMR, Dual Modular Redundancy (DMR), and even-Parity schemes (XOR-based encode/decode). For the Parity schemes, different bit-slice widths (3, 10, and 16 bits) were explored. Each SM produces an error signal, and these are OR-ed together into a single top-level output.

For the validation experiments, specific safety thresholds (e.g., ASIL-B) were selected as targets for the hardened circuits. The same SMs used during exploration were inserted during implementation, and validation was performed by both comparing PPA trends and performing quantitative correlation.

All experiments were executed on Synopsys' customer server farm.

VI. EXPERIMENTAL RESULTS

A. SPFM COVERAGE ANALYSIS

We used the safety static analyzer on the RTL design to estimate the SPFM coverage of each unhardened circuit. This analysis provided an SPFM coverage loss value for each register. Registers were ranked by SPFM coverage loss, and the cumulative sum of this metric is shown in Figure 2 for the RISC-V cores. The figure illustrates the estimated SPFM coverage (y-axis) achieved when hardening a certain percentage of registers (x-axis), using the reported ranking. Due to this ordering, the slope of the curve progressively decreases (i.e., achieving higher SPFM coverage requires hardening progressively more registers).

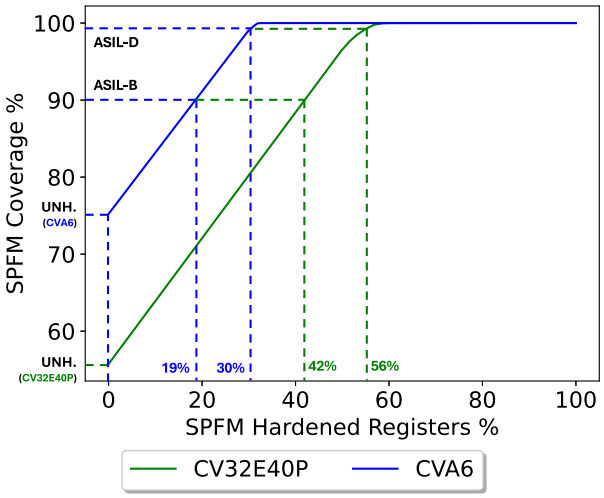


FIGURE 2. Cumulative sum of the SPFM coverage loss metrics derived by the RTL safety static analyzer for the CV32E40P and CVA6 processors.

Since more than 45% of the registers in the CV32E40P core, and more than 29% in the CVA6 core, exhibit similar loss values, the left portion of each curve in Figure 2 is nearly linear. The remaining registers have progressively more negligible loss values and contribute minimally to SPFM coverage when hardened. The estimated SPFM coverage of the unhardened circuits was 55.74% for the CV32E40P and 75.22% for the CVA6, corresponding to the leftmost points on each curve. Using this representation, designers can estimate the number of registers that must be hardened to reach a target SPFM level. The figure shows that for

the CV32E40P, achieving ASIL-B and ASIL-D compliance requires hardening approximately 42% and 56% of the registers, respectively, while for the CVA6, approximately 19% and 30% are required. While this type of analysis provides a high-level overview of the hardening complexity, it does not address the impact of SM insertion on circuit metrics.

B. HARDENING EXPLORATION

This section details the experimental analysis conducted to assess multiple hardening strategies and their effects on PPA, as summarized below. The exploration phase aims to characterize the PPA tradeoffs associated with different SMs before committing to a full implementation. This phase provides early, design-accurate predictions that guide RTL designers in selecting the most suitable hardening strategy for each register or subsystem.

Performance is evaluated primarily through Total Negative Slack (TNS) and Number of Violating Paths (NVP), both of which capture the extent to which timing constraints are violated across the design. TNS reflects the aggregate timing deficit, while NVP indicates how widespread the violations are. Worst Negative Slack (WNS) is computed but not emphasized, as it reflects only the single worst path and does not provide a representative view of global timing behavior. Power refers to the total estimated power consumption, while area corresponds to the total footprint after synthesis. Together, these metrics provide a comprehensive view of how hardening affects design PPA.

The complexity of an SM refers to the amount of additional logic it introduces, particularly registers. Mechanisms such as Parity add minimal overhead, while DMR, and especially TMR, introduce significantly more circuitry. Higher SM complexity generally increases logic count, routing congestion, and optimization effort. However, with increasing SM complexity (e.g., from Parity-16 to Parity-3), better protection against double-bit errors is provided, along with increased LFM and, in some cases (e.g., TMR), error correction capability. As complexity increases, higher area and power consumption are expected, along with more challenging timing closure.

The exploration phase is precisely what enables designers to study these trade-offs early, before committing to a final hardening strategy to be implemented. While many results align with intuition (e.g., more complex SMs generally degrade timing and increase area), several behaviors are counterintuitive. These deviations underscore the importance of an exploration phase based on accurate predictions rather than assumptions.

To evaluate the impact of different SMs on PPA, we conducted experiments in which all selected registers were hardened uniformly using a single SM type. Although real designs typically employ a mix of SMs selectively, this controlled setup allows us to isolate the effect of each mechanism and validate the trade-offs it introduces. The same methodology applies when different SMs are combined in a single design.

All PPA values are normalized using standard scaling (shown in Equation (1)) to enable consistent comparison across mechanisms and ASIL targets.

$$z = \frac{x - u}{s} \tag{1}$$

In Equation (1), z is the normalized value, x is the original value, u is the mean of all values in the corresponding dataset, and s is the standard deviation.

Several perspectives are analyzed across different designs and ASIL targets:

- CV32E40P hardening comparison for ASIL-B and ASIL-D (Figure 3)
- CV32E40P and CVA6 hardening trends for ASIL-D (Figure 4)
- ISCAS89 benchmarks hardening trends for ASIL-C and ASIL-D (Figure 5).

1) CV32E40P HARDENING COMPARISON FOR ASIL-B AND ASIL-D

Figure 3 reports normalized PPA metrics for the CV32E40P design across ASIL-B and ASIL-D targets. Most SMs behave consistently across ASIL-B and ASIL-D, with a few exceptions. TNS and NVP trends are nearly identical across all SMs and ASIL levels. Power and area trends reveal additional nuances. While area generally increases with SM complexity, some cases show slight decreases (e.g., from Parity-16 to Parity-3 in ASIL-B). Power shows peaks with Parity-16 and TMR implementations, while remaining lower for DMR. These outliers may arise from the synthesis tool applying more aggressive optimizations when handling more complex SMs, occasionally finding better local minima.

2) CV32E40P AND CVA6 HARDENING TRENDS FOR ASIL-D

Figure 4 compares CV32E40P and CVA6 hardening for an ASIL-D target. For CVA6, DMR and Parity schemes improve timing relative to the unhardened design, while area and power scale more predictably with SM complexity. Unlike the CV32E40P, the CVA6 exhibits a linear power trend, potentially due to its larger size, which likely results in more consistent optimization effort from the tool.

3) ISCAS89 BENCHMARKS HARDENING TRENDS FOR ASIL-C AND ASIL-D

Figure 5 reports normalized PPA data for the ISCAS89 benchmarks under ASIL-C and ASIL-D targets. ASIL-B is omitted because most circuits already meet its SPFM requirement.

The ASIL-C exploration on the ISCAS89 benchmark suite, shown in Figure 5a, reveals largely consistent behavior across the circuits, with unhardened implementations generally achieving the best TNS. Parity-16 consistently degrades TNS compared to the unhardened version, with s38417 being an exception that shows minimal improvement. In most

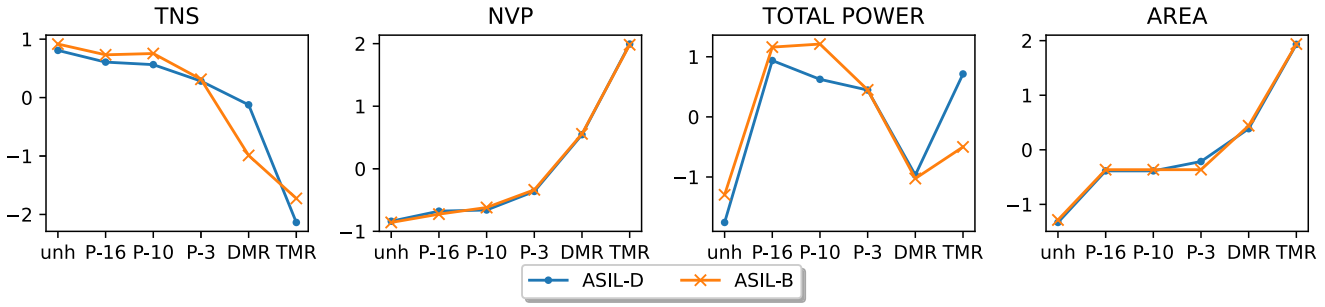


FIGURE 3. Exploration CV32E40P PPA trends comparison for different hardening levels (y-axis normalized metric values, x-axis hardening type).

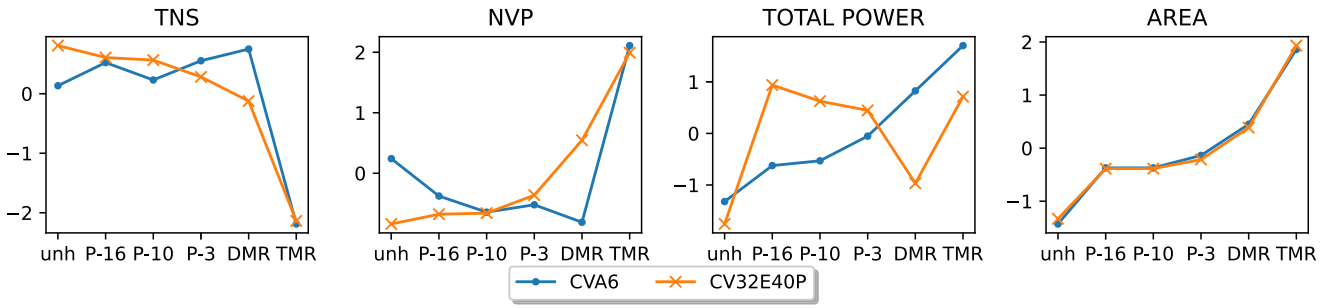


FIGURE 4. Comparison between CV32E40P and CVA6 exploration ASIL-D PPA trends (y-axis normalized metric values, x-axis hardening type).

cases, Parity-10 shows better or comparable TNS relative to Parity-16. Parity-3 produces more heterogeneous outcomes: some circuits, such as s9234, experience improved timing relative to Parity-10 and Parity-16, whereas others degrade, reflecting the sensitivity of smaller designs to optimization heuristics. DMR, in contrast, provides better TNS than Parity implementations on s9234 and s38584. TMR degrades TNS the most across all benchmarks. NVP trends follow a predictable progression, increasing with SM complexity and showing particularly steep growth for TMR. The most notable exception is s38417, which shows a peak in the Parity-16 implementation. Power consumption also tends to rise with more complex SMs, though s38417 again behaves atypically, with Parity-16 and Parity-10 requiring less power than even the unhardened version. Area overheads scale proportionally with SM complexity for most circuits, with s9234 being the primary exception. These observations collectively illustrate how design size and structure influence the interaction between SM insertion and synthesis optimizations.

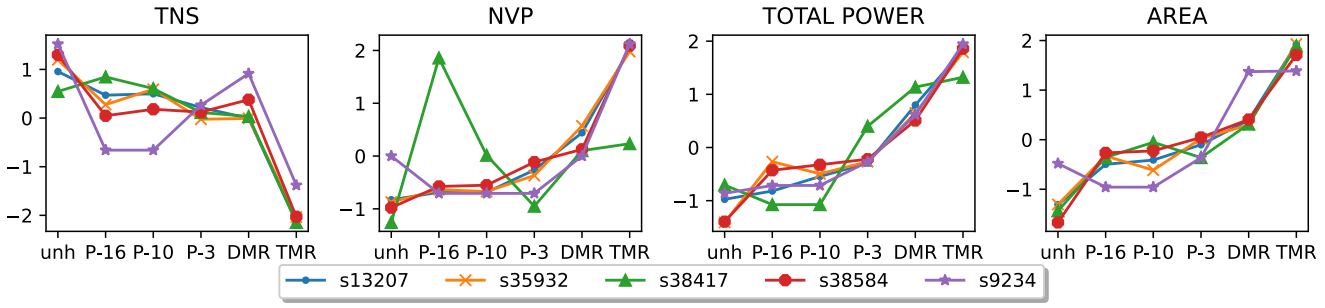
Figure 5b shows more stable and predictable trends under ASIL-D constraints. TNS behavior is consistent across nearly all circuits. Parity-10 shows better TNS than Parity-16 for most implementations. NVP increases monotonically with SM complexity, with only minor deviations. Power and area exhibit a quadratic growth pattern beginning with Parity-10 and becoming especially pronounced for modular redundancy schemes, particularly TMR. Compared to ASIL-C, the ASIL-D results show stronger alignment with conceptual expectations, largely because the tighter constraints trigger higher tool effort and more uniform optimization behavior. Overall, the ASIL-D exploration highlights how increased safety requirements lead to more predictable PPA trends, while also emphasizing the substantial overhead introduced by high-complexity SMs.

4) KEY REMARKS

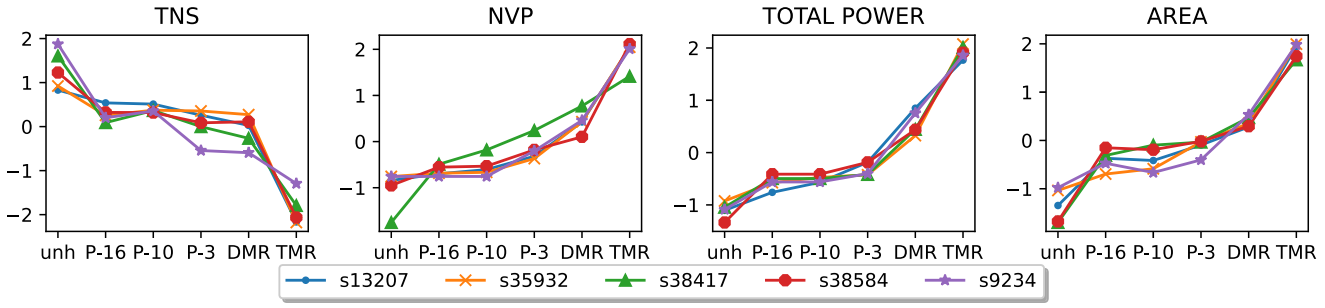
Overall, across all designs and ASIL targets, SM complexity correlates with PPA degradation, but the relationship is not strictly predictable. Tool heuristics, design size, and structural characteristics introduce non-obvious behaviors that cannot be reliably predicted without exploration. The exploration phase therefore plays a critical role in identifying optimal hardening strategies and avoiding over- or under-protection. The next subsection demonstrates how these exploration-based predictions translate into actual implementation results, validating the accuracy and usefulness of the early evaluation methodology.

C. HARDENING IMPLEMENTATION

To validate the accuracy of the exploration flow, we implemented hardened versions of the circuits for each evaluated safety mechanism and compared the resulting PPA with the exploration estimates across different ASIL targets. Specifically, validation was performed for the following configurations:



(a) ASIL-C PPA trends



(b) ASIL-D PPA trends

FIGURE 5. Comparison between ISCAS89 benchmarks ASIL-C and ASIL-D PPA trends (y-axis normalized metric values, x-axis hardening type).

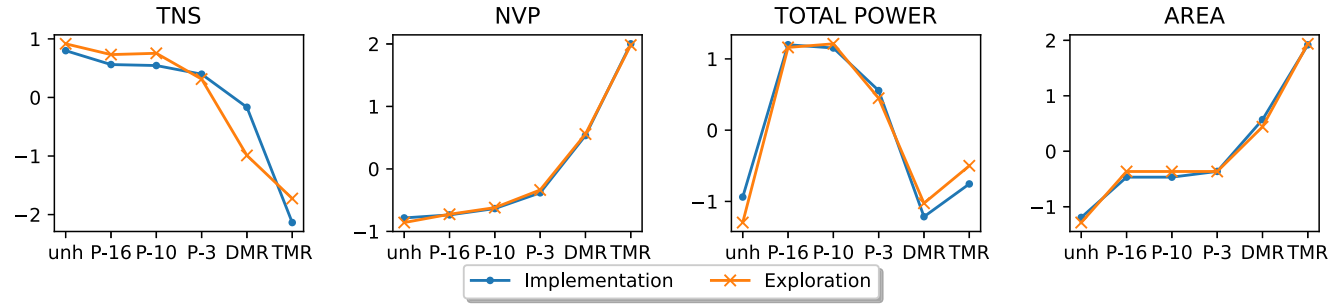


FIGURE 6. Comparison between CV32E40P exploration and implementation ASIL-B PPA trends (y-axis normalized metric values, x-axis hardening type).

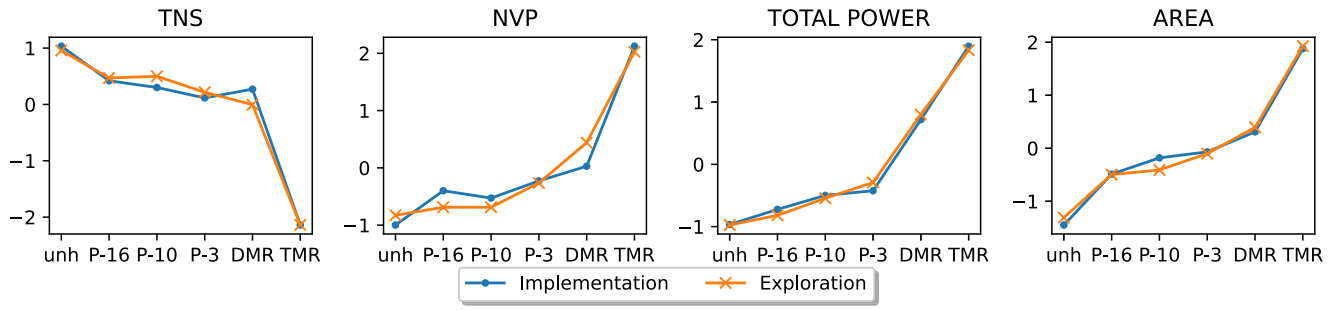
- CV32E40P exploration and implementation PPA trends for an ASIL-B target (Figure 6)
- ISCAS89 benchmarks exploration and implementation PPA trends for ASIL-C and ASIL-D targets (Figure Figures 7 to 10)

1) CV32E40P EXPLORATION AND IMPLEMENTATION PPA TRENDS FOR AN ASIL-B TARGET

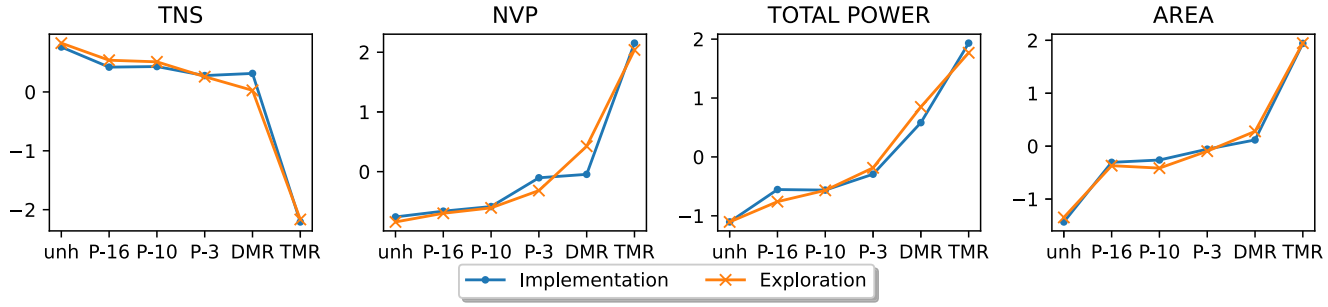
A normalized comparison between exploration and implementation results is shown in Figure 6. The two trends align closely across most of the design space, confirming the reliability of the exploration flow for architecture-level hardening decisions. A few deviations are worth noting. For Parity-10, the exploration shows a slightly increasing TNS trend, whereas the implementation shows a mild decrease. Despite this inversion, the normalized TNS and NVP behaviors remain consistent with the patterns observed in Figure 3, supporting the hypothesis that tool effort differences influence these variations. Power trends for Parity-10 also diverge slightly, rising during exploration and falling during implementation, although both slopes remain small. Area trends for Parity-3 differ between the two flows but converge toward similar normalized values.

2) ISCAS89 BENCHMARKS EXPLORATION AND IMPLEMENTATION PPA TRENDS FOR ASIL-C AND ASIL-D TARGETS

The same validation methodology was applied to the ISCAS89 benchmark circuits for ASIL-C and ASIL-D hardening. Figures 7 to 10 report results for s13207, s38417, s35932, and s38584. Overall, the correlation between exploration and implementation is high across nearly all scenarios, with only a few localized discrepancies. For example, in the ASIL-C DMR implementation of s35932,



(a) ASIL-C PPA trends



(b) ASIL-D PPA trends

FIGURE 7. Comparison between s13207 benchmark exploration and implementation ASIL-C and ASIL-D PPA trends (y-axis normalized metric values, x-axis hardening type).

the TNS is one relative point higher than predicted, mirroring the deviation observed in the Parity-3 case. NVP differences of similar magnitude appear in the DMR and Parity-10 scenarios. Significant NVP differences are observed in the s38417 results. TNS and NVP together provide a clear picture of the overall timing closure status. NVP alone, however, can be highly sensitive when paths shift from marginally failing to marginally passing. Hence, while NVP is needed to complete the picture of overall timing distribution, outliers in its behavior are secondary when TNS tracks well between exploration and implementation, especially when TNS is close to zero, as in the Parity-3 configuration.

Power trends also show minor mismatches: the unhardened implementation consumes more power than predicted, and the Parity-16 trend reverses direction relative to exploration. A comparable inversion appears in the area results for Parity-10. Under ASIL-D, the DMR implementation exhibits fewer NVPs than predicted, and the Parity-3 area trend increases during exploration but decreases slightly during implementation. In the s38584 benchmark, the ASIL-C DMR implementation shows a decreasing TNS trend, opposite to the slight increase predicted during exploration. A similar inversion appears in the ASIL-D NVP trend for the same circuit.

3) QUANTITATIVE CORRELATION

To increase the confidence level in the results, in addition to visual trends comparisons, quantitative correlation has been approached.

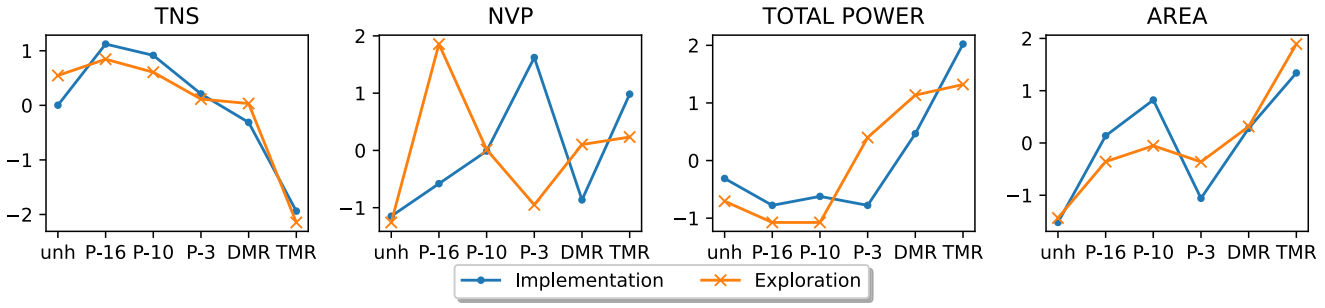
Quantitative correlation has been computed through the Pearson coefficient, between exploration and implementation flows, across the different inserted SMs and the analyzed PPA metrics. Results are shown in Table 3, reporting both the r coefficients and the p-values.

Every single r value is above 0.975, with most exceeding 0.99. This means the exploration and implementation flows produce PPA trends that are nearly identical. The lowest value is Parity-16 NVP at 0.9759, which is still very strong.

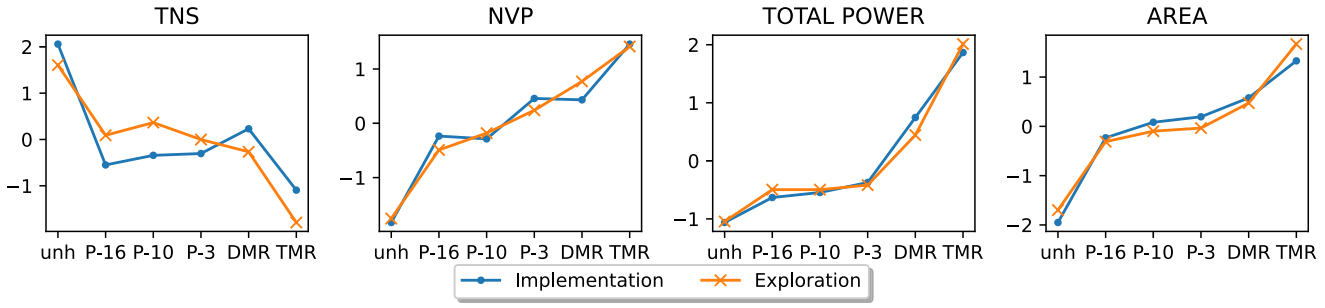
The p-values are all extremely small, ranging from 10e-05 (for the unhardened, which has fewer data points) down to 10e-17. This means the correlations are highly statistically significant and the probability of observing such strong correlations by chance is essentially infinitesimal. Even the largest p-value (3.13e-05 for Unhardened NVP) is well below any conventional significance threshold.

4) KEY REMARKS

The discrepancies observed in validation trends are subtle and expected. By design, the exploration flow does not apply the fine-grained optimizations available during full implementation. As a result, small trend inversions may occur in isolated cases. Nevertheless, the strong quantitative correlation demonstrates that the exploration flow provides a dependable foundation for early-stage hardening analysis and accurately anticipates the behavior of final implemented designs, thus confirming that the safety strategy selected during exploration remains valid after implementation.



(a) ASIL-C PPA trends



(b) ASIL-D PPA trends

FIGURE 8. Comparison between s38417 benchmark exploration and implementation ASIL-C and ASIL-D PPA trends (y-axis normalized metric values, x-axis hardening type).

TABLE 3. Pearson correlation coefficients (r) and p-values between exploration and implementation flows across safety mechanisms and PPA metrics.

	TNS		NVP		Power		Area	
Mechanism	r	p	r	p	r	p	r	p
Unhardened	0.9956	2.49e−06	0.9878	3.13e−05	0.9975	6.22e−07	0.9951	3.25e−06
Parity-3	0.9968	1.85e−13	0.9898	1.13e−10	0.9980	1.66e−14	0.9991	1.38e−16
Parity-10	0.9950	2.43e−12	0.9889	1.83e−10	0.9972	9.54e−14	0.9985	3.36e−15
Parity-16	0.9945	4.01e−12	0.9759	1.27e−08	0.9972	9.25e−14	0.9993	6.73e−17
DMR	0.9942	5.47e−12	0.9873	3.93e−10	0.9974	6.10e−14	0.9912	5.32e−11
TMR	0.9977	3.15e−14	0.9895	1.40e−10	0.9981	1.31e−14	0.9985	2.60e−15

D. RUNTIME IMPROVEMENT

The runtimes achieved with the exploration flow are promising. The percentage improvements are shown in Table 4. Results are computed as the mean percentage improvement across all scenarios (i.e., unhardened and each SM type) for each design.

The benchmark results show a 33.3% improvement for smaller circuits, while for larger ones, improvement reaches 60.0%. The RISC-V cores exhibit improvements of 42.2% for the CV32E40P and 46.9% for the CVA6.

Given the highly iterative nature of the digital design process, these improvements compound over time, providing significant TTM reduction.

TABLE 4. Exploration vs. Implementation mean runtime improvement (MRTI).

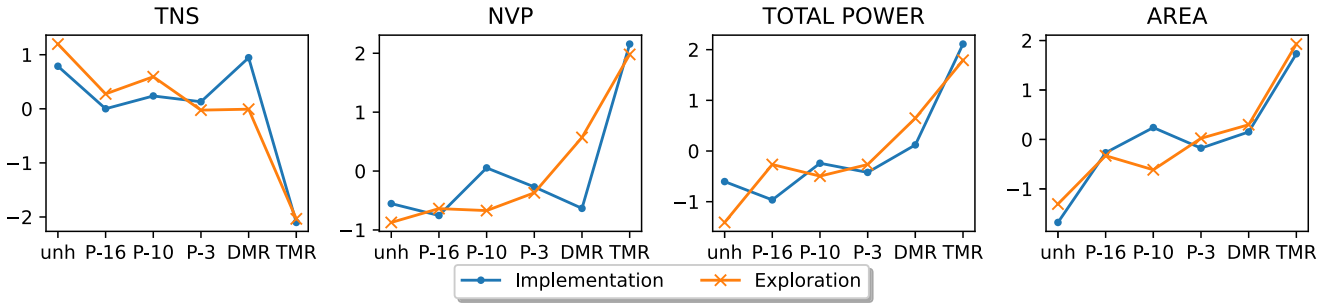
Design	MRTI [%]
s9234	33.3
s13207	33.3
s35932	60.0
s38417	60.0
s38584	50.0
cv32e40p	42.2
cva6	46.9

E. TARGET SPFM COVERAGE VALIDATION

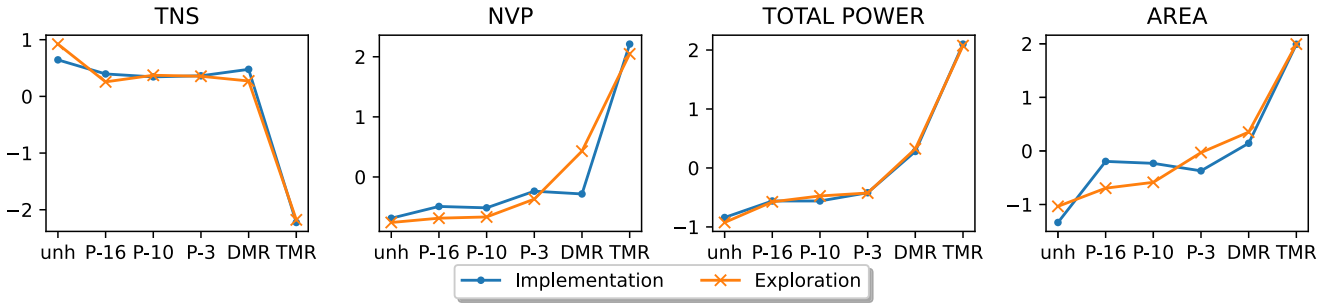
Finally, after comparing and validating the hardening descriptor SSF files produced during exploration against those produced during implementation, the safety static analyzer was run on the post-implementation netlists. The SSF-informed analysis of the netlists confirms that the target SPFM coverage was achieved.

VII. CONCLUSION

This work addresses the growing need for an optimized digital design methodology capable of handling increasing complexity and stringent Functional Safety (FuSa) requirements. The proposed safety-aware design flow enables early, concurrent exploration of Power, Performance, and

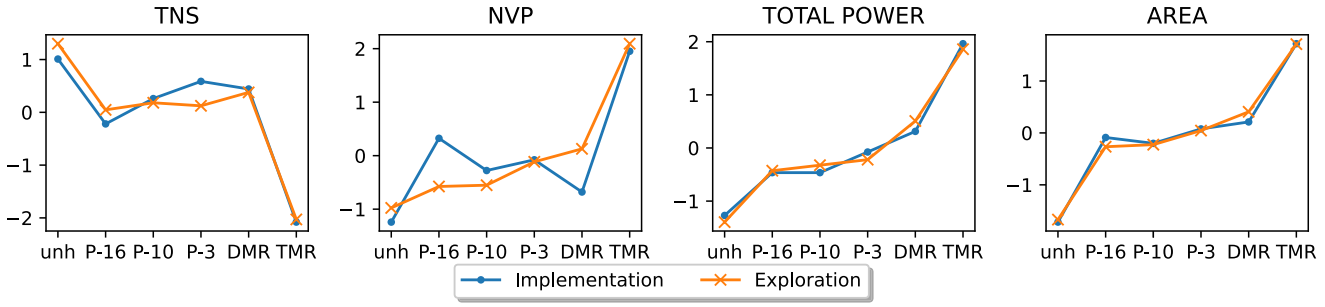


(a) ASIL-C PPA trends

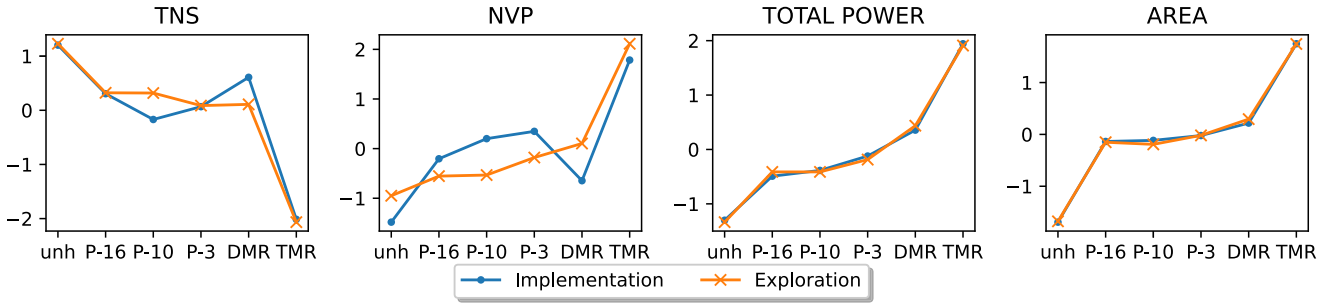


(b) ASIL-D PPA trends

FIGURE 9. Comparison between s35932 benchmark exploration and implementation ASIL-C and ASIL-D PPA trends (y-axis normalized metric values, x-axis hardening type).



(a) ASIL-C PPA trends



(b) ASIL-D PPA trends

FIGURE 10. Comparison between s38584 benchmark exploration and implementation ASIL-C and ASIL-D PPA trends (y-axis normalized metric values, x-axis hardening type).

Area (PPA) metrics alongside safety and reliability considerations, enabling RTL designers to evaluate hardening strategies with significantly greater efficiency. The complete end-to-end flow is driven by safety file descriptors (SSF), which provide a unified mechanism for specifying and propagating safety intent throughout the design process.

Additionally, the impact of Safety Mechanisms (SMs) on key safety metrics is quantified using a static safety analyzer operating directly at the RTL level, providing early and accurate visibility into FuSa compliance.

The exploration phase demonstrated substantial runtime reductions, up to 60% for larger designs, significantly accelerating iterative refinement and reducing Time-To-Market (TTM). Designers were able to study the PPA implications of different SMs, revealing both expected and counterintuitive trade-offs that underscore the value of early-stage analysis. The methodology produced PPA predictions that closely matched implementation results, giving designers confidence that decisions made during exploration remain valid through final implementation. Furthermore, automated synthesis targeting ASIL SPFM requirements enabled rapid experimentation with diverse protection strategies while maintaining a clear understanding of their PPA impact.

By shifting critical decision-making to the exploration stage, the flow minimizes costly late-stage adjustments and supports a more streamlined, safety-aware development process. Future work will focus on adaptive hardening approaches that tailor SMs to module-level characteristics and on a unified flow that empowers RTL designers to guide SM definition and insertion more directly. Overall, this methodology represents a significant step toward faster, more predictable development of safety-critical semiconductor designs.

REFERENCES

[1] *Cerebras's Wafer-Scale Engine 3: The Largest Chip Ever Built*. Accessed: Mar. 3, 2025. [Online]. Available: https://cerebras.ai/press-release/cerebras-announces-third-generation-wafer-scale-engine

[2] *IEC 61508-1:2010-Functional Safety Of Selectrical/Electronic/Programmable Electronic Safety-Related Systems*, Standard IEC 61508-1, 2010.

[3] *ISO 26262-1:2018-Road Vehicles-functional Safety*, Standard ISO 26262-1, 2018.

[4] *RTL Architect: Parallel RTL Exploration With Unparalleled Accuracy-white Paper*, Synopsys, Sunnyvale, CA, USA, 2021.

[5] D. Niu, D. Zhang, Y. Cao, Z. Jin, C. Wang, Y. Dong, and C. Sun, "A novel image-graph heterogeneous fusion framework for static IR drop prediction," in *Proc. 62nd ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2025, pp. 1–7.

[6] H. Ren, S. Nath, Y. Zhang, H. Chen, and M. Liu, "Why are graph neural networks effective for EDA problems?" in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Oct. 2022, pp. 1–8.

[7] Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, "PowerNet: Transferable dynamic IR drop estimation via maximum convolutional neural network," in *Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2020, pp. 13–18.

[8] Y. Zhao, Z. Chai, X. Jiang, Y. Lin, R. Wang, and R. Huang, "PDNNet: PDN-aware GNN-CNN heterogeneous network for dynamic IR drop prediction," in *Proc. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, Jun. 2025, vol. 44, no. 6, pp. 2253–2263.

[9] H. Che, D. Niu, Z. Zang, Y. Cao, and X. Chen, "Ed-drap: Encoder–decoder deep residual attention prediction network for radar echoes," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 1004705.

[10] A. Nardi and A. Armato, "Functional safety methodologies for automotive applications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 970–975.

[11] 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.

[12] V. P. Nelson, "Fault-tolerant computing: Fundamental concepts," *Computer*, vol. 23, no. 7, pp. 19–25, Jul. 1990.

[13] R. Hentschke, F. Marques, F. Lima, L. Carro, A. Susin, and R. Reis, "Analyzing area and performance penalty of protecting different digital modules with Hamming code and triple modular redundancy," in *Proc. 15th Symp. Integr. Circuits Syst. Design*, 2002, pp. 95–100.

[14] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 6, pp. 2874–2878, Dec. 1996.

[15] H. Wang, X. Dai, Y. M. Y. Ibrahim, H. Sun, I. Nofal, L. Cai, G. Guo, Z. Shen, and L. Chen, "A layout-based rad-hard DICE flip-flop design," *J. Electron. Test.*, vol. 35, no. 1, pp. 111–117, Feb. 2019.

[16] I. Polian and J. P. Hayes, "Selective hardening: Toward cost-effective error tolerance," *IEEE Design Test Comput.*, vol. 28, no. 3, pp. 54–63, May 2011.

[17] P. C. Murley and G. R. Srinivasan, "Soft-error Monte Carlo modeling program, SEMM," *IBM J. Res. Develop.*, vol. 40, no. 1, pp. 109–118, Jan. 1996.

[18] R. A. Weller, M. H. Mendenhall, R. A. Reed, R. D. Schrimpf, K. M. Warren, B. D. Sierawski, and L. W. Massengill, "Monte Carlo simulation of single event effects," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 4, pp. 1726–1746, Aug. 2010.

[19] G. Asadi and M. B. Tahoori, "An accurate SER estimation method based on propagation probability," in *Proc. Design, Autom. Test Eur.*, 2005, pp. 306–307.

[20] R. R. Rao, K. Chopra, D. Blaauw, and D. Sylvester, "An efficient static algorithm for computing the soft error rates of combinational circuits," in *Proc. Design Autom. Test Eur. Conf.*, Mar. 2006, pp. 1–6.

[21] R. R. Rao, K. Chopra, D. T. Blaauw, and D. M. Sylvester, "Computing the soft error rate of a combinational logic circuit using parameterized descriptors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 3, pp. 468–479, Mar. 2007.

[22] D. Alexandrescu, E. Costenaro, and M. Nicolaidis, "A practical approach to single event transients analysis for highly complex designs," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, Oct. 2011, pp. 155–163.

[23] A. Evans, D. Alexandrescu, E. Costenaro, and L. Chen, "Hierarchical RTL-based combinatorial SER estimation," in *Proc. IEEE 19th Int. On-Line Test. Symp. (IOLTS)*, Jul. 2013, pp. 139–144.

[24] M. Maniatakos, M. Michael, C. Tirumurti, and Y. Makris, "Revisiting vulnerability analysis in modern microprocessors," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2664–2674, Sep. 2015.

[25] G. I. Paliaroutis, P. Tsoumanis, N. Evmorfopoulos, G. Dimitriou, and G. I. Stamoulis, "Placement-based SER estimation in the presence of multiple faults in combinational logic," in *Proc. 27th Int. Symp. Power Timing Model., Optim. Simul. (PATMOS)*, Sep. 2017, pp. 1–6.

[26] M. Anglada, R. Canal, J. L. Aragón, and A. González, "Fast and accurate SER estimation for large combinational blocks in early stages of the design," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 3, pp. 427–440, Jul. 2021.

[27] G.-I. Paliaroutis, P. Tsoumanis, D. Garyfallou, A. Vagenas, N. Evmorfopoulos, and G. Stamoulis, "Accurate soft error rate evaluation using event-driven dynamic timing analysis," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2023, pp. 1–6.

[28] F. Wang and Y. Xie, "Soft error rate analysis for combinational logic using an accurate electrical masking model," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 1, pp. 137–146, Jan. 2011.

[29] W. Shida, T. Min, Z. Hongwei, M. Bo, and S. Yi, "Evaluation of single-event upset in FinFET device," in *Proc. 5th Int. Conf. Radiat. Effects Electron. Devices (ICREED)*, May 2023, pp. 1–7.

[30] A. Nardi, D. Kingston, D. Johnson, M. Ayache, G. Kanawati, and J. Forey, "Statistical analysis of architectural vulnerability factor for soft errors," in *Proc. GOMACTech*, 2026.

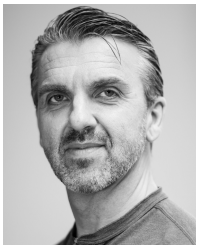
[31] C. G. Zoellin, H.-J. Wunderlich, I. Polian, and B. Becker, "Selective hardening in early design steps," in *Proc. 13th Eur. Test Symp.*, 2008, pp. 185–190.

[32] I. Polian, S. M. Reddy, and B. Becker, "Scalable calculation of logical masking effects for selective hardening against soft errors," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Apr. 2008, pp. 257–262.

[33] A. K. Nieuwland, S. Jasarevic, and G. Jerin, "Combinational logic soft error analysis and protection," in *Proc. 12th IEEE Int. Line Test. Symp. (IOLTS06)*, Jul. 2006, pp. 99–104.

[34] O. Ruano, J. A. Maestro, and P. Reviriego, "A methodology for automatic insertion of selective TMR in digital circuits affected by SEUs," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 4, pp. 2091–2102, Aug. 2009.
[35] J. E. R. Condia, P. Rech, F. F. dos Santos, L. Carrot, and M. S. Reorda, "Protecting GPU's microarchitectural vulnerabilities via effective selective hardening," in *Proc. IEEE 27th Int. Symp. Line Test. Robust Syst. Design (IOLTS)*, Jun. 2021, pp. 1–7.
[36] *Synopsys: Safety Specification Format (SSF)*. Accessed: Jan. 28, 2025. [Online]. Available: https://www.synopsys.com/automotive/safety-\specification-format.html
[37] *OpenHW's Group: CV32E40P CPU Core*. Accessed: Jan. 30, 2025. [Online]. Available: https://github.com/openhwgroup/cv32e40p
[38] *OpenHW's Group: CVA6 CPU Core*. Accessed: Jan. 30, 2025. [Online]. Available: https://github.com/openhwgroup/cva6
[39] *ISCAS89's Benchmarks*. Accessed: Nov. 13, 2025. [Online]. Available: https://web.archive.org/web/20211104172\900/https://people.engr.ncsu.edu/brglez/CBL/benchma\rks/ISCAS89
[40] *Synopsys's SAED32: 32nm Implementation Library*. Accessed: Sep. 22, 2025. [Online]. Available: https://www.synopsys.com/apps/protected/university\members.html#libraries

MICHELANGELO BARTOLOMUCCI (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering, specialized in embedded systems, from the Politecnico di Torino, Italy. He is currently pursuing the Ph.D. degree. His M.Sc. thesis was focused on automating functional safety solutions for achieving given standards compliance in digital circuits. In his Ph.D. thesis, he's continuing the work, focusing on next generation methodologies for functional safety and is performed in collaboration between the Politecnico di Torino and Synopsys.



DAVID KINGSTON received the B.Sc. degree in communications engineering from the University of Kent, U.K. He is a Chartered Engineer (C.Eng.) and a Member of the Institution of Engineering and Technology (MIET). He is a Solutions Architect working as part of the Advanced Platform Solutions Team, Synopsys, focusing on EDA solutions for functional safety and reliability. After graduating, he held positions as an ASIC Design Engineer with Mentor Graphics and an Independent ASIC Design Consultancy (GF Micro), also with Magma Design Automation working as a Product Engineer, for implementation tools.

TEO CUPAIUOLO received the M.S. degree in electrical engineering from the Polytechnic University of Turin, Italy, in 2006. He is currently a Solutions Engineer with the Advanced Platform Solutions Team, Synopsys, focusing on EDA solutions for functional safety. Prior to that, he was an Application Engineer with Yogitech for functional safety tools for semiconductors. Earlier in his career, he held various roles, including a Test Engineer with Pirelli, and a RTL Designer and a Verification Engineer with STMicroelectronics.

ALESSANDRA NARDI received the M.S. and Ph.D. degrees in electrical engineering from the University of Padua, Italy. She is the Executive Director of the Advanced Platforms Solutions Team, Synopsys, where she oversees a cross-functional team focused on the development and implementation of electronic design automation (EDA) solutions for functional safety, security, and reliability. After her Postdoctoral Researcher with the University of California, Berkeley, she held various research and development lead positions with Magma Design Automation, Cadence and Synopsys spanning statistical static timing analysis, library characterization, power integrity, functional safety, and reliability.

RICCARDO CANTORO (Member, IEEE) received the M.S. and Ph.D. degrees in computer engineering from the Politecnico di Torino, Turin, Italy, in 2013 and 2017, respectively. He is currently an Associate Professor with the Department of Computer Engineering, Politecnico di Torino. His research interests include software-based functional testing of SoC and memories, and machine learning applied to test and diagnosis.