Early FMEDA at RTL for Functional Safety: Correlating RTL Metrics to GLS for Accurate Architectural Analysis

Myungkyoon Yim, Principal Researcher – Hyundai Korea (mk.yim@hyundai.com)
Vedant Garg, Solutions Architect – Synopsys Austin (vedantg@synopsys.com)
Buyong UM, Senior Researcher - Hyundai Korea (buyong.um@hyundai.com)
Kyle Kim, Application Engineer – Synopsys Korea (chunsung@synopsys.com)
Liz Song, Account Management – Synopsys Korea (jieun@synopsys.com)

Abstract

Ensuring functional safety in automotive electronic systems, as mandated by ISO 26262, requires comprehensive FMEDA throughout the hardware lifecycle. Traditional FMEDA approaches rely on gate-level simulations and post-layout data, limiting opportunities for early architectural exploration and design optimization. This paper introduces a methodology for conducting FMEDA at the Register Transfer Level (RTL), enabling design teams to assess safety metrics, perform fault injections, and analyze diagnostic coverage much earlier in the development process. Early identification of single-point and latent faults allows designers to address safety risks and enhance safety mechanisms prior to implementation. [3]

To validate RTL-based FMEDA accuracy, a correlation framework is presented that aligns RTL results with those from gate-level simulations. This analysis demonstrates the relationship between early and post-synthesis safety metrics and highlights the impact of logic optimizations and timing effects on diagnostic coverage. The methodology supports architectural analysis, safety mechanism partitioning, and hierarchy-level safety metric evaluation. It provides practical tools for designers and safety managers to achieve robust early safety assessment and efficient ISO 26262 compliance, with applications extending to automotive, aerospace, and industrial domains. [2] [3]

Keywords: ISO 26262, Functional Safety, FMEDA, RTL Fault Injection, Diagnostic Coverage, ASIL, Safety Metrics

1. Introduction

1.1 Motivation

Ensuring *functional safety* in today's automotive semiconductor systems is not just a regulatory obligation, it's a design necessity. Modern vehicles integrate hundreds of electronic control units (ECUs) and complex System-on-Chip (SoC) architectures, each required to demonstrate compliance with ISO 26262. The standard emphasizes *quantitative evaluation of random hardware failures*, which forms the foundation of safety metrics such as the Single Point Fault Metric (SPFM), Latent Fault Metric (LFM), and the Probabilistic Metric for Random Hardware Failures (PMHF).

Traditionally, the quantitative analysis for these metrics is performed late in the design cycle using Gate-Level Simulation (GLS) or post-layout safety analysis. While accurate, this approach introduces significant constraints:

- It delays visibility into safety vulnerabilities until after synthesis and place-and-route.
- It limits design teams' ability to optimize safety mechanisms when architectural flexibility is highest.
- It increases verification turnaround time, as fault injections at gate-level require extensive computational resources.

The lack of early safety visibility often results in *reactive design iterations* discovering safety weaknesses when the design is already implementation-bound. This reactive model conflicts with the ISO 26262 objective of *"building safety in by design"*.

This work is motivated by a simple but powerful question: **Can safety metrics and fault detectability be meaningfully quantified at the RTL stage, well before final synthesis while maintaining accuracy and traceability to GLS results?**

If this correlation is established, engineers can identify safety-critical weaknesses early, make data-driven architectural choices, and drastically shorten the time to reach safety sign-off.

1.2 Objectives and Contributions

This work presents RTL centric FMEDA framework that performs fault analysis and safety metric derivation early in the design cycle, while maintaining structural correlation with gate-level implementations. Although all analysis and reporting are conducted at the RTL abstraction, the methodology internally performs a fast synthesis step to extract a structural view of the design. This step captures gate-level relationships such as *fan-in/fan-out cones, sequential boundaries,* and *combinational connectivity* without waiting for full place-and-route. The result is a hybrid model that retains RTL visibility for the user but incorporates sufficient structural detail to accurately represent the underlying logic behavior.

This hybridization allows fault sites to be mapped more precisely and ensures that the diagnostic coverage, SPFM, and LFM metrics derived at RTL remain *quantitatively traceable* to gate-level FMEDA. In effect, the end user performs FMEDA entirely from the RTL environment, while the toolchain transparently maintains synthesis-informed accuracy.
The key objectives and contributions of this work are:

1. **Early Safety Quantification at RTL:** Enable ISO 26262 compliant FMEDA execution at the RTL stage using fault injection and classification, leveraging synthesis-derived structural context for accurate fault propagation analysis. [4] [5]
2. **Structural Correlation Between RTL and GLS:** Introduce a mapping layer that correlates RTL fault targets to synthesized netlist nodes, enabling metric-level comparison and correlation between early and final FMEDA results.
3. **Architectural Decision Enablement:** Provide data-driven safety metrics (SPFM, LFM, DC) that allow architects to optimize redundancy, parity coverage, ECC configurations, and monitoring strategies while the design is still malleable.
4. **Automation and Time Efficiency:** Demonstrate a fully automated RTL FMEDA flow that can execute fault campaigns in parallel, achieving substantial runtime savings and an estimated *30% reduction* in end-to-end safety verification closure. [1]

2. Traditional Methodology

Gate level methodology operates on final synthesized netlists representing the tapeout-ready design where all logic optimizations, hierarchy flattening, and timing constraints have already been applied. Although this process yields accurate post-implementation data, it significantly limits flexibility for early safety exploration and architectural optimization.

2.1 Gate-Level Netlist Generation

In the traditional flow, the starting point for FMEDA creation is the final gate-level netlist, which has been fully synthesized and timing-closed. This netlist contains all flattened logic cells and interconnections as generated by synthesis and optimization tools.

Table 1: Snippet of flattened nodes and area

Flattened Node (Netlist)	Digital_Area (μm^2)
U3254/AOI22_X1	4.8
U3255/INV_X2	1.5
U3256/NAND3_X1	3.6
U3257/BUF_X1	1.2
U3258/DFFR_X1	4.3
U3259/OR2_X1	2.4

At this stage, the synthesis tool provides detailed information about each instance, including its cell type, hierarchical name, and digital area as shown in Table 1. However, since the design hierarchy is flattened for timing and optimization purposes, the mapping between these netlist nodes and the original RTL hierarchy is effectively lost. While this data provides accurate implementation metrics, it is purely *flat* lacking any connection to architectural intent or functional hierarchy. This is where the complexity of generating a meaningful FMEDA begins.

2.2 Generating FMEDA

In a flattened gate-level environment, generating FMEDA requires manual reconstruction of design hierarchy and safety associations. Each netlist node must be traced back to its originating RTL or functional block, which can be extremely challenging since synthesis tools may rename or merge signals, remove redundant logic, or retime registers. Analysts often resort to manual schematic exploration, opening hierarchical views in EDA tools (e.g., Verdi) and visually correlating logic cones to their corresponding functional modules. This process is time-consuming, error-prone, and highly dependent on the engineer's understanding of the synthesis transformations.

For example, a register originally located in **top.cpu.alu.reg_status[3]** at RTL may appear as multiple distributed flip-flops **(U3258/DFFR_X1, U4110/DFFR_X1)** at gate level due to retiming or optimization.

- The analyst must manually correlate these new flattened nodes back to the ALU functional block to assign the appropriate failure modes and diagnostic coverage.
- Further complexity arises when defining Safety Mechanisms (SMs). At this stage, hardware-based safety mechanisms such as parity checkers, error correction units, and watchdogs are separated from mission logic—but any missing protection cannot be easily added.
- The FMEDA must explicitly classify the failure modes associated with both the safety mechanism and the unprotected mission logic, often through tedious cross-referencing between schematics and fault simulation results.

As a result, hierarchy reconstruction, fault association, and coverage quantification become a highly manual and iterative process, often requiring weeks of effort for large SoCs.

2.3 Validation by Fault Injection

Once the FMEDA table is completed, fault injection campaigns are carried out on the gate-level netlist to validate the effectiveness of the implemented safety mechanisms. During this phase, faults such as stuck-at-0, stuck-at-1, or transient events are simulated to assess whether the safety mechanisms can detect or mitigate these issues. If diagnostic gaps are uncovered, it is often too late for hardware changes, as the netlist represents the final tapeout-ready design. As a result, any unresolved safety weaknesses must be addressed at the software or system level, which adds complexity and may introduce performance trade-offs. For example, if fault injections reveal that a data bus lacks parity protection, hardware modifications would require an expensive design re-spin. Instead, teams are typically forced to implement software-based integrity checks or periodic memory read-back operations, which increase operational latency and compromise the goal of hardware-driven safety. This challenge highlights the importance of early-stage safety visibility an area specifically addressed by the RTL-based FMEDA methodology discussed in the following section. [5]

3. RTL FMEDA Methodology

The proposed RTL FMEDA methodology establishes an early, data-driven framework for quantifying safety metrics and validating safety mechanisms using fault injections. It bridges the abstraction gap between RTL design intent and gate-level implementation accuracy through a combination of lightweight synthesis, hierarchical modeling, automated failure mode distribution, and fault injection validation.

3.1 RTL Abstraction through lightweight quick Synthesis

Table 2: Structural data extract from our methodology

Cell_Type	Digital_Area (μm²)	Transistor Count
DFF_X1	4.2	20
NAND2_X1	2.5	10
NOR3_X2	3.8	14
INV_X1	1.1	6
XOR2_X1	3.2	12
AOI21_X2	4.5	16

This flow is anchored by lightweight synthesis, an EDA-driven process that converts RTL into a structural abstraction comprising primitive cells, flip-flops, and combinational elements. Unlike full synthesis, which includes optimization and placement, lightweight synthesis focuses on extracting key structural details such as cell composition, digital area, and transistor count for each module without altering the design as shown in Table 2. These quantitative attributes serve as the foundation for the FMEDA model, with each cell annotated by its digital area (in μm²) and equivalent transistor count, enabling accurate scaling of hardware failure

probabilities. This approach maintains implementation awareness while keeping the analysis at the RTL level. The resulting data can be exported in CSV or XML formats, streamlining downstream automation and mapping these metrics into the RTL hierarchy. [4]

3.2 Creating RTL Hierarchy Based FMEDA Data

Once the structural data is available, each RTL block or sub-block is modeled as a hierarchical node containing its respective cell composition. The lightweight synthesis data is aggregated per hierarchy, summing up the digital area or transistor count for all constituent cells. Mathematically, the cumulative metric for each RTL module M can be expressed as:

$$\mathbf{Metric(M) = i\sum n(Ni \times Ai)}$$

where N_i represents the number of occurrences of cell type i in the hierarchy and A_i denotes its corresponding digital area or transistor count. This process allows a quantitative FMEDA view to be constructed across the design hierarchy as shown in Table 3 below:

Table 3: RTL hierarchy based FMEDA data

RTL_Hierarchy	Total_Cells	Aggregated_Area (µm²)	Total_Transistors
top.cpu.alu	22,450	65,800	290,000
top.cpu.decoder	11,320	34,200	155,000
top.memory.controller	14,970	42,750	195,000
top.io.interface	8,540	21,900	99,800

This aggregation ensures that every RTL block has a quantifiable representation of its hardware complexity, forming the baseline for **Failure Rate (λ)** and **Failure Mode (FM)** estimation. [3] [4]

3.3 Validation through Fault Injection

To validate FMEDA assumptions and the coverage of safety mechanisms, fault injection is performed on safety-relevant failure modes identified by the hierarchical FMEDA model. Each injected fault such as stuck-at-1/0, transient, or control-path failure simulates realistic hardware defects, allowing the system's safety mechanisms to be evaluated for their ability to detect or mitigate these faults. The validation process includes automatic generation of fault lists and simulation setups, execution of injection campaigns at the RTL level using fault simulators (such as VCZ01X), and logging and classification of results into Detected, Latent, or Residual faults.

The outcomes from fault injection are fed back into the FMEDA model to refine detection coverage and update SPFM and LFM metrics. This **closed-loop approach** enables targeted architectural feedback, allowing undetected faults to be traced directly to specific RTL modules so that safety mechanisms can be improved early in the design. As a result, the methodology **supports continuous safety enhancement**, bridging architectural analysis with verification closure. By transforming traditional post-synthesis safety analysis into an early, iterative process, the RTL FMEDA methodology leverages quick synthesis-based structural mapping, hierarchical FMEDA generation, uniform failure distribution, and RTL-level fault injection for both speed and traceability. This empowers teams to validate diagnostic coverage and safety metrics in compliance with ISO 26262 well before gate-level implementation, significantly **reducing the likelihood of costly late-stage design changes**.

4. Correlation between RTL FMEDA and Traditional FMEDA

To assess the accuracy of RTL-based FMEDA, we quantitatively compared its results with traditional gate-level FMEDA performed on the final synthesized netlist. This comparison aimed to evaluate how well RTL-derived safety metrics predict post-synthesis reliability and coverage.

The overall Failure In Time (FIT) and Probabilistic Metric for Random Hardware Failures (PMHF) for the selected IP block are summarized below. While the absolute FIT value differed noticeably between RTL and gate-level analyses primarily due to synthesis optimizations and area reduction the PMHF remained consistently correlated. These differences are largely attributed to logic optimizations during synthesis, such as gate merging and retiming, which lower transistor count and, in turn, reduce the net failure rate at the gate level. Importantly, the distribution of faults and effectiveness of safety mechanisms remained consistent, confirming that early RTL-based FIT estimates reliably predict overall reliability trends. A strong correlation was also observed for SPFM (Single Point Fault Metric) and LFM (Latent Fault Metric), as illustrated in Figure 1. The RTL FMEDA reported an SPFM of 76.0%, while the gate-level FMEDA measured 71.18%, resulting in a deviation of approximately 6.34%. Both approaches converged at 99% LFM, demonstrating excellent alignment in identifying latent fault paths.

Metric	FIT (Failures in Time)	PMHF (FIT)	SPFM (%)	LFM (%)
RTL FMEDA	0.1805	0.07821	76.0	99.0
Gate-Level FMEDA	0.09933	0.06012	71.18	99.0
Delta	0.08117	0.01809	6.34%	0.0%

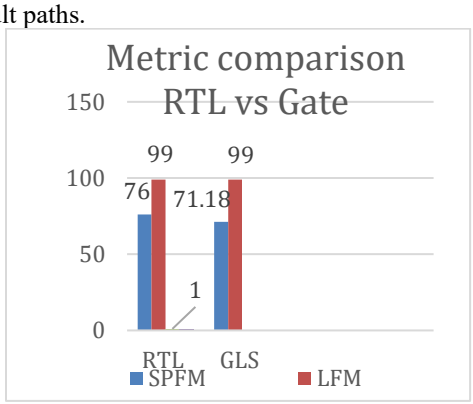

Figure 1: Comparison of safety metrics between RTL and Gate-Level FMEDA

These results validate that architectural fault coverage assessed at the RTL stage is largely maintained through synthesis, with the minor SPFM delta primarily caused by small timing-driven changes affecting checker activation windows during gate-level simulation.

The analysis demonstrates that RTL FMEDA offers highly **predictive reliability**, accurately estimating post-synthesis safety metrics such as SPFM and LFM with minimal deviation, despite the area and logic reductions introduced by synthesis. By preserving design hierarchy, the RTL flow ensures clear **architectural traceability** of safety mechanisms to their respective logic blocks, greatly simplifying cause-and-effect analysis and reducing FMEDA setup and debugging time compared to gate-level approaches. Most notably, the strong correlation between RTL and gate-level safety metrics confirms that RTL FMEDA can effectively provide **early safety closure** that can replace late-stage gate-level simulations for preliminary safety certification, enabling earlier design closure and reducing overall verification effort by up to 35%. Although absolute FIT and PMHF values vary with implementation, the consistent and traceable diagnostic coverage metrics derived at RTL validate it as a robust and efficient early-stage methodology for safety-critical IP evaluation under ISO 26262.

5. Case Studies

5.1 RISCV Open Core (For model)

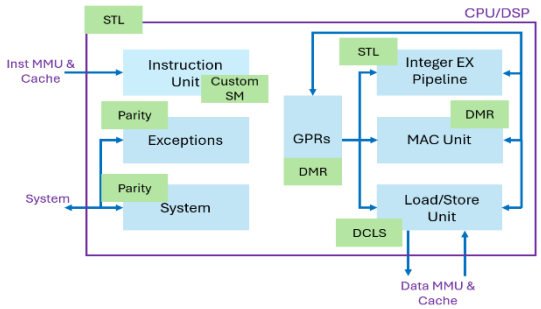

Figure 2: Block diagram of CPU with modelled SM

Simple Hierarchy Name	Digital_Area
or1200_alu	8580.00
or1200_rf/rf_a	7413.00
or1200_mult_mac/or1200_gmultp2_...	3971.00
or1200_sprs	2875.00
or1200_except	2765.00
or1200_mult_mac	2677.00
or1200_genpc	1850.00
or1200_ctrl	1153.00
or1200_operandmuxes	768.00
or1200_wbmux	761.00
or1200_if	590.00
top	499.00

Figure 3: RTL abstracted hierarchy with Digital area

For our experiments, we selected an open-source RISC-V processor core as the test platform. Safety mechanisms including parity checks, watchdog timers, and lockstep redundancy were implemented across various functional blocks at the RTL level to evaluate their impact on early safety metrics as shown in Figure 2. These same mechanisms were consistently integrated into the gate-level netlist to enable direct comparison between RTL and gate-level results. Additionally, a software test library was developed and applied at both abstraction levels to facilitate detection of specific failure modes.

Name	SPFM	LFM	PMHF	PMHF Distribution	FD	λ	λ_{nSR}	λ_{nS}	λ_{PVSG}	In-use DC(K_{RF})	λ_{RF}
SoC	77.23%	-	8.167E-02	100.00%	100.00%	1.923E-01	0.000E+00	1.385E-01	1.385E-01	77.23%	1.809E-02
or1200	77.23%	-	8.167E-02	100.00%	100.00%	1.923E-01	0.000E+00	1.385E-01	1.385E-01	77.23%	1.809E-02
or_CPU	76.00%	-	7.821E-02	95.76%	93.87%	1.805E-01	0.000E+00	1.311E-01	1.311E-01	76.00%	1.802E-02
or_DC	99.00%	-	1.606E-03	1.97%	2.62%	5.043E-03	0.000E+00	3.413E-03	3.413E-03	99.00%	3.413E-05
or_IC	99.00%	-	1.192E-03	1.46%	2.77%	5.332E-03	0.000E+00	2.539E-03	2.539E-03	99.00%	2.539E-05
or_PIC	99.00%	-	6.624E-04	0.81%	0.74%	1.424E-03	0.000E+00	1.424E-03	1.424E-03	99.00%	1.424E-05

Figure 4: Overall safety metrics at RTL showing CPU SPFM at 76%

At the RTL stage, we performed lightweight synthesis to extract hierarchical data such as module boundaries and digital area as shown in Figure 3, which were used to construct the FMEDA. Our initial analysis showed that the Single Point Fault Metric (SPFM) for the CPU block reached 76%, indicating robust diagnostic coverage as shown in Figure 4. After implementing the modified RTL design and running synthesis, optimization techniques such as logic minimization and resource sharing reduced the digital area across most blocks as shown in Figure 5. This reduction led to a change in safety metrics; specifically, the CPU core's SPFM decreased to 71.18% post-synthesis, a reduction of 6.34% compared to the RTL estimate as shown in Figure 6. This difference highlights the impact of synthesis optimizations on diagnostic coverage and the need to correlate RTL and gate-level safety assessments.

Simple Hierarchy Name	Digital_Area
or1200_rf/rf_a	4439.00
or1200_alu	3173.00
or1200_mult_mac/or1200_gmultp2_...	2370.00
or1200_except	1675.00
or1200_sprs	1632.00
or1200_mult_mac	1599.00
or1200_genpc	1050.00
or1200_ctrl	776.00
or1200_wbmux	456.00
or1200_operandmuxes	423.00
or1200_lsu_shadow	334.00
or1200_lsu	334.00
or1200_if	324.00

Figure 5: Gate hierarchy with Digital area

Name	SPFM	LFM	PMHF	PMHF Distribution	FD	λ	λ_{nSR}	λ_{nS}	λ_{PVSG}	In-use DC(K_{RF})	λ_{RF}
SoC	73.16%	-	6.356E-02	100.00%	100.00%	1.067E-01	0.000E+00	1.041E-01	1.041E-01	73.16%	1.794E-02
or1200_cpu	71.18%	-	6.012E-02	94.59%	93.08%	9.933E-02	0.000E+00	9.667E-02	9.667E-02	71.18%	1.787E-02
or1200_dc_top	99.00%	-	1.465E-03	2.30%	2.94%	3.140E-03	0.000E+00	3.140E-03	3.140E-03	99.00%	3.140E-05
or1200_ic_top	99.00%	-	1.588E-03	2.50%	3.19%	3.399E-03	0.000E+00	3.399E-03	3.399E-03	99.00%	3.399E-05
or1200_pic	99.00%	-	3.852E-04	0.61%	0.79%	8.408E-04	0.000E+00	8.408E-04	8.408E-04	99.00%	8.408E-06

Figure 6: Overall safety metrics at Gate showing CPU SPFM at 71.18%

To validate our findings, we conducted fault injection experiments at the gate-level netlist. A random sample set of 12,500 stuck-at faults was randomly injected across various blocks, with results analyzed at both pre- and post-verification stages. Key metrics obtained were **Fsafe**, representing the ratio of faults deemed safe, and **DC**, representing fault coverage as shown in Figure 7. By comparing FMEDA-derived safety metrics after synthesis to the empirical fault injection results, we established a quantitative correlation between gate-level FMEDA and early RTL-derived metrics as shown in Table 4. This comparative analysis demonstrates the importance of early safety assessment and ongoing validation throughout the design flow.

```
                                                    Total
---------------------------------------------------------------------
Number of Faults:                                   12500

Untestable Faults:                                     59
  Untestable Unused                        UU          46
  Untestable Tied                          UT          13

Testable Faults:                                    12441

Status Groups -------------------------------------------------------
  Assumed Dangerous Unobserved             AU         197
  Dangerous Detected                       DD        8813
  Dangerous Not Detected                   DN        1126
  Unselected                               NG        2305
  Untestable                               UG          59

Coverage ------------------------------------------------------------
  Fsafe                                               0.98%
  DC                                                 70.50%
  Errors                                                 0
```

Figure 7: Fault injection results on netlist

Table 4: Correlation and validated results RTL vs GATE (pre & post verification)

Metric	**RTL Abstraction** (Proposed)	**GLS w/o verification** (Traditional Est)	**GLS w Verification** (Traditional Actual)	**Correlation Est** (Traditional Est vs Traditional Actual)	**Correlation Actual** (Traditional Actual vs Proposed)	**Overall Prediction Accuracy**
SPFM	76%	71.18%	70.78% (With Fsafe)	70.78 / 71.18 = 0.995 (99.5%)	70.78 / 76 = 0.932 (93.2%)	(99.5+93.2) /2 = 96.35%
LFM	99%	99%	-	-	-	100%

5.2 Automotive SoC (Actual)

For this case study, we focused on a representative IP block from our Automotive System-on-Chip (SoC portfolio). To gather meaningful insights, we carried out a lightweight synthesis, which allowed us to extract detailed hierarchical abstraction data. This data formed the basis for constructing Failure Modes, Effects, and Diagnostic Analysis (FMEDA). Figure 8 is a figurative example of abstraction on RTL, example showing GPIO block.

Figure 9 represents an early intermediate result for the FMEDA built on RTL data. As this IP is currently in production, we expect some variation. As part of our evaluation, we calculated the Single Point Fault Metric (SPFM) for the IP. The results showed an SPFM of 96.49%, demonstrating a high degree of diagnostic coverage at the RTL stage. [5]

Simple Hierarchy Name	Digital_Area
iomux_aud_top/iomux_aud_sfr	3202.00
iomux_aud_top/iomux_aud	1173.00
iomux_aud_top/iomon_aud_gpio0	477.00
iomux_aud_top/iomon_aud_gpio10	477.00
iomux_aud_top/iomon_aud_gpio11	477.00
iomux_aud_top/iomon_aud_gpio12	477.00
iomux_aud_top/iomon_aud_gpio13	477.00
iomux_aud_top/iomon_aud_gpio14	477.00
iomux_aud_top/iomon_aud_gpio15	477.00
iomux_aud_top/iomon_aud_gpio1	477.00
iomux_aud_top/iomon_aud_gpio2	477.00
iomux_aud_top/iomon_aud_gpio3	477.00

Figure 8: Automotive SoC RTL abstraction with Digital area

This finding highlights the robustness of our design and its suitability for safety-critical automotive applications.

Name	SPFM	LFM	PMHF	PMHF Distribution	FD	λ	λ_{nSR}	λ_{nS}	λ_{PVSG}	In-use	λ_{RF}
IPDEV	96.49%	-	9.587E-03	100.00%	100.00%	4.008E-01	2.382E-02	9.252E-02	1.371E-02	72.37%	3.787E-03
DMA_Wrapper	100.00%	-	8.858E-05	0.92%	13.90%	5.572E-02	4.589E-03	1.885E-02	0.000E+00	NaN%	0.000E+00
GPIO_Controller	94.33%	-	6.284E-03	65.55%	21.80%	8.737E-02	1.923E-02	6.683E-02	1.371E-02	72.37%	3.787E-03
OTP_Wrapper	NaN%	-	0.000E+00	0.00%	0.00%	0.000E+00	0.000E+00	0.000E+00	0.000E+00	NaN%	0.000E+00
PWM	NaN%	-	0.000E+00	0.00%	62.45%	2.503E-01	0.000E+00	0.000E+00	0.000E+00	NaN%	0.000E+00
Top_FCCU	100.00%	-	3.214E-03	33.53%	1.85%	7.405E-03	0.000E+00	6.839E-03	0.000E+00	NaN%	0.000E+00

Figure 9: Automotive SoC – IP Safety metric view with SPFM at 96.49%

5.3 Correlation Analysis

The consistency of Single Point Fault Metric (SPFM) values across design stages can be measured by calculating the ratio of post-verification to pre-verification SPFM at the gate level. In this study, the ratio is approximately 0.995, indicating an excellent match between the diagnostic coverage predicted by gate-level FMEDA and the actual coverage validated through fault injection. The difference between these stages is minimal, with a delta of just 0.4%.

Furthermore, comparing gate-level post-verification SPFM to RTL abstraction SPFM yields a correlation coefficient ratio of approximately $\rho = 0.93$. This demonstrates that diagnostic coverage is only slightly reduced after synthesis and implementation optimizations. The SPFM shows strong correlation across all design stages, confirming that both hardware and software safety mechanisms modeled at RTL are effectively preserved through synthesis. Similarly, the Latent Fault Metric (LFM) calculated by FMEDA shows robust alignment between RTL and gate-level results. These outcomes validate RTL FMEDA as a reliable predictor of final safety performance, allowing design teams to gain early confidence in diagnostic coverage and fault classification, thereby minimizing uncertainty before physical implementation as shown in Tabel 5 below.

Table 5: SPFM Correlation Ratios Across Design Stages

Comparison Stage	SPFM Ratio (ρ)	Delta (%)	Interpretation
Correlation Est (Traditional Est vs Traditional Actual)	0.995	0.4%	High consistency between FMEDA and fault injection
Correlation Actual (Traditional Actual vs Proposed)	0.93	7%	Slight decrease after synthesis and optimizations

6. Conclusion

This work demonstrates a robust and experimentally validated RTL-based FMEDA methodology that enables early, quantitative functional safety assessment in accordance with ISO 26262 Part 5 and Part 11. By integrating fault injection, fault classification, and comprehensive safety metric derivation at the RTL stage, the approach provides design teams with a predictive and structured understanding of safety risks and diagnostic coverage well before physical implementation.

The methodology's use of lightweight synthesis abstraction effectively bridges the gap between design intent and hardware realization, allowing accurate extraction of digital area and transistor data for early FMEDA analysis. Automated data aggregation and targeted fault injections facilitate reliable calculation of key metrics such as SPFM and LFM, supporting informed architectural decisions during initial development phases.

A major contribution of this work is the correlation framework that consistently aligns RTL-based FMEDA results with gate-level outcomes. Case studies demonstrated a high correlation coefficient ($\rho = 0.93$), confirming that RTL-derived safety metrics are strong predictors of post-synthesis diagnostic coverage. The minimal differences observed are within engineering tolerances, validating the methodology's reliability for safety-critical applications. Overall, this RTL-based FMEDA methodology empowers design and verification teams with **Predictive Reliability** for early safety evaluation, **Architectural Traceability** for robust decision-making, and **Early Safety Closure** for efficient ISO 26262 compliance—laying the foundation for **true shift left on using RTL based FMEDA methodology**.

The complete RTL FMEDA workflow including quick synthesis tools, open-source FMEDA resources, fault injection setup/reports, and correlation analysis is available upon request. For access or more information, please contact Vedant Garg at vedantg@synopsys.com.

7. References & Acronyms

[1] Methodology for Efficient closure to Fault injection, Dvcon Europe 2022
[2] ISO 26262 Part 5: Product development at the hardware level, Second edition 2018-12
[3] ISO 26262 Part 11: Guidelines on application of ISO26262 to semiconductors, Second edition 2018-12
[4] Rambus RT-640 road to ISO26262 certification, Rambus white paper
[5] Complex Safety Mechanisms Need Interoperability for Validation and Close Loop for Final Metrics, Protecting Safety and Security, DVCon US 2023

Acronym	Full Form / Description
FMEDA	Failure Modes, Effects and Diagnostic Analysis
FSIM	Fault Simulation
Fsafe	Safe Fault (Faults with no safety impact)
LFM	Latent Fault Metric
SM	Safety Mechanism
SPFM	Single Point Fault Metric