Gulroz Singh$^1$, Ankit Hegde$^2$, and Vaibhav Kumar$^1$

$^1$MCU/MPU Engineering, NXP Semiconductors Austin
$^2$MCU/MPU Engineering, NXP Semiconductors Munich

May 13, 2024
In the rapidly advancing landscape of System-on-Chips (SoCs), achieving appropriate levels of functional safety compliance has become a fundamental concern. With SoCs serving critical roles in applications such as automotive, industrial, and consumer electronics, a unified approach to navigate different functional safety standards can help save time and speed up time to market. By reusing the fundamental principles of ISO 26262 and IEC 61508, this article presents a unified framework tailored specifically for advanced multi-domain SoCs. This framework seeks to streamline safety compliance efforts while enhancing overall system safety resilience for various applications.

Keywords—Functional Safety, ISO 26262, IEC 61508, Fault Tree Analysis, Failure Modes Effects and Diagnostics Analysis, Dependent Failure Analysis

I. INTRODUCTION

In the rapidly advancing landscape of System-on-Chips (SoCs), achieving appropriate levels of functional safety compliance has become a fundamental concern. With SoCs serving critical roles in applications such as automotive, industrial, and consumer electronics, a unified approach to navigate different functional safety standards can help save time and speed up time to market. By reusing the fundamental principles of ISO 26262 and IEC 61508, this article presents a unified framework tailored specifically for advanced multi-domain SoCs. This framework seeks to streamline safety compliance efforts while enhancing overall system safety resilience for various applications.

Safety critical systems in automotive, industrial and consumer electronics applications require appropriate levels of functional safety compliance to manage system risks. Multi-domain System-on-Chips (SoCs) require adherence to multiple functional safety standards during the development process. Following different safety standards in silos can be challenging from time-to-market perspective and may even lead to redundant work. The unified functional safety framework presents a model-based safety critical SoC development framework which incorporates SoC design, safety analysis, SoC verification and assessment methodology. The proposed framework addresses the issue of effective reuse of design and safety artifacts that can streamline development processes for SoCs that need compliance with ISO 26262 and IEC 61508.

Section II of the article details the unified functional safety framework with illustrations.

II. UNIFIED FUNCTIONAL SAFETY FRAMEWORK

The framework focusses on four main aspects of SoC development flow from a functional safety perspective as shown in Fig 2. SoC safety architecture and requirements form the first important aspect of the framework. We then present our effective reuse approach for safety analysis methods such as FTA, DFA and FMEDA. Effective use and reuse of pre-silicon verification methods such as Fault Injection Simulation for verifying diagnostic coverage of safety mechanisms forms the third important aspect of the framework. We conclude the discussion with effective techniques for functional safety assessment for ISO 26262 and IEC 61508.

A. Safety Architecture

In order to build an optimal SoC architecture that can support both automotive and industrial safety applications, it is required to define a Safety Concept/Architecture that is applicable for both these industries. The first important step is to analyze the automotive and industrial safety use cases applicable for the SoC which helps in deriving a common safety concept/architecture for the SOC. One of the ways to have a common concept that works for both standards is the Safety Monitoring concept where one part of the SOC executes a high end QM function and other part of the SOC executes a monitoring function of the QM parts or external system in a safe way. Monitoring function ensures a generalized safety concept that can be applied to multiple domains without conflicts.

Taking an example of an SoC that has applications in automotive and industrial industries the use cases considered can be as follows:
• Automotive use-cases: Digital cockpit, Infotainment system, emerging AI/ML applications
• Industrial: Industrial machine automation, smart home, Building and energy control

With the above SoC use cases in mind, a system application can be imagined as a combination of QM and Safety application with the necessary isolation as shown in Fig 1. As shown, the Safety application monitors the execution of the QM application required for the target ASIL/SIL of the system. For industrial use cases, QM application can be Main (EUC) application and Safety application can be the Safety monitoring of the EUC. For automotive, QM application can be the infotainment system function and the Safety application can be the Safety monitoring of the infotainment display. This type of concept in Automotive is called ASIL decomposition as per ISO26262:2018 Part 9 Clause 5.

As the SoC architecture modelling process can be iterative with multiple abstractions and views, model based systems architecture is the recommended method. Model based design enables traceable and consistent view of the SoC architecture and allows linkage to requirements at different levels and safety analysis methods. Figure 3 shows the integration of model based architecture with the abovementioned artifacts.

B. Safety Analysis

Safety analysis methods such as Fault tree analysis, Failure Modes, Effects and Diagnostics and Dependent Failure analysis are critical to verify the safety concept of a SoC. Both ISO 26262 and IEC 61508 recommends to perform inductive, deductive and common cause failure analysis to exhaustively verify the safety concept. However, the exact tables in the standard for the abovementioned safety analysis have minor differences. The authors believe that the fundamentals of the guidelines provided are quite similar in the standards with minor adjustments required for specific methods.

Fault Tree Analysis (FTA) is a deductive analysis safety analysis method which is used often to verify the completeness and accuracy of the safety concept. As part of the Unified functional safety framework, we recommend to use functional approach to FTA where high level functions and malfunctions are identified first followed by failure modes of the individual IPs. To best align these high level SoC functions, the framework proposes to create a functional dependency net for the SoC functions and malfunctions which clarifies the flow and dependencies of the functions of the SoC. This prework streamlines the creation of the fault tree and helps in reducing the effort spent in brainstorming the FTA flow. This technique also streamlines cut sets that are created from the fault tree making the analysis of these cut sets less error prone.

According to ISO 26262, Dependent Failure Analysis includes common cause failure analysis and cascading failure analysis. However, IEC 61508 provides guidance on estimating CCFs employing the beta factor method. In this case, even though the guidance from the standards deviate, the core concepts for common cause analysis still remains the same. The framework recommends to use specific templates to harmonize analysis from both standards to avoid rework and costly effort during safety concept assessments for both standards. The framework also recommends connecting cut sets from the FTA and the cut set analysis for Dependent Failure Analysis which ensures effective change management and traceability. These above mentioned proposals are in line with the overall integrated model based safety analysis approach mentioned earlier.

Failure Modes, Effects and Diagnostics analysis (FMEDA) is a popular quantitative inductive safety analysis method used to calculate failure metrics according to safety standards. ISO 26262 and IEC 61508 have different metrics guidance and hence different methodology recommended for these two standards. However, the common safety architecture proposed earlier becomes a glue between FMEDA guidance from the two standards. Model based architecture allows the safety analyst to connect relevant failure attributes to the model elements such as failure modes, failure rates and failure rate distribution. This allows the analyst to use the common model as a starting point for FMEDA which eventually complies to both ISO 26262 and IEC 61508. The details of the abovementioned approach is explained through Fig 3.

C. Verification

Fault injection verification is recommended to be one of the techniques performed during pre-silicon verification to correctly verify the diagnostic coverage of the safety mechanisms used in a system. Fault injection verification introduces faults into the test object during runtime using specialized test interface which allows set diagnostic, detection, and observation points. To optimally define the diagnostic and observation points, inputs from architects, designers along with verification engineers becomes quite important to accurately perform a fault injection campaign. It can also be used to verify safety mechanisms for their correctness, interconnectivity, and verifying specific corner cases.

There are two methodologies of Fault injection that can be performed to improve evidence of diagnostic coverage beyond expert judgement. Directed Fault injection: In a simulation environment, a series of test cases are performed to analyze the effectiveness of safety mechanisms to detect failure modes. Faults are injected to represent different failure modes during the safe operational modes of the HW. The decision of the sufficient coverage of the test cases is reviewed between the verification team and the Safety team. If the test involved a SW Error injection, a SW Safety engineer is involved to review that the test case is relevant to the behavior of the SW. This test provides a qualitative result of the effectiveness of Safety mechanisms to detect every failure mode claimed in the safety analysis. The other kind is called Random Fault injection in which a representative number of faults are injected in an automated way in the design at the HW subpart level. The injected faults are classified and simulated, and the results are used for quantitative analysis. This Fault injection will happen at RTL level and stimulates fault in functional blocks like
register, adders, multiplexers. Irrespective of the type of fault injection used, a pre-simulation step with Formal fault analysis reduces the fault that need to be injected using static and mathematical analysis. Consequently, faults are injected in the RTL model using the following (Stuck at Faults, Bridge faults, Open Faults, Stuck-open faults). This ensures timely results from fault injection simulation campaigns.

Fault injection campaigns require important resources, advance tooling and deep expertise. Therefore, the tactical approach to this plan is to reduce the scope to the most critical element in the design that require fault injection, and rely on requirement based testing and analysis for less critical Safety mechanisms.

III. CONCLUSION

Multi-domain SoCs necessitates a unified functional safety framework that integrates principles of both ISO 26262 and IEC61508. Our contribution harmonizes the two standards which enables scalability and adaptability of the functional safety processes and artifacts.

REFERENCES