Design-For-Safety For Automotive IC Design: Challenges And Opportunities

Alessandra Nardi
Cadence Design Systems, Inc
San Jose,CA 95134,USA
anardi@cadence.com

Samir Camdzic
Integrated Power Management, Texas Instruments
samir@ti.com

Antonino Armato
Cadence Design Systems, Inc
San Jose,CA 95134,USA
armato@cadence.com

Francesco Lertora
Cadence Design Systems, Inc
San Jose,CA 95134,USA
lertoraf@cadence.com

***Abstract* — As the automotive industry marches towards higher level of autonomous driving, the design supply chain is shaping up to support a great level of complexity and new requirements, such as Functional Safety, that have not been traditionally part of the design/verification/implementation flow. The full automation supported in EDA (Electronic Design Automation) tools for traditional metrics has not yet reached maturity for the new safety metrics and is a green field of innovation very much needed to holistically enable the development of semiconductors for autonomous driving. This paper introduces the requirements to Design-For-Safety, it presents some of the challenges and opportunities for flow automation, and it also reviews commonalities and differences between the Digital and Analog/Mixed-Signal (A/MS) flows.**

Keywords — Functional safety, Automotive, Semiconductors, Digital, Mixed-signal, Design-For-Safety, Verification, Design

I. Introduction

ADAS (Advanced Driver Assistance System) features such as Automatic Cruise Control, Lane Departure Warning, and Automatic Parking are already in production in the market and several players in the automotive industry are targeting to reach technology readiness for level 5 "mind off" [10] well within a decade. This growing demand for complex SoC for safety-critical sensor-fusion control applications poses new challenges to the semiconductor industry, both from the architectural and process point of view, to address the dependability requirements for autonomous driving cars.

Integrated circuits, both digital and A/MS, are the core technology needed to support the autonomous driving paradigm "Sense-Decide-Actuate": they provide the sensor technology (e.g. radar, lidar, camera), the processing brains (e.g. CPU, GPU) and the power circuits for actuation (e.g. motor drives) [1]. Their development requires the strict application of Functional Safety (FS) defined as the "absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical and/or electronic systems" [5]. For non-safety critical applications, EDA already provides a holistic approach to develop and verify integrated circuits and some of the FS specific features are also now available for use. However, several aspects of the complete flow are still to be developed or integrated. Moreover, while the state of the art on the FS analysis of digital devices is extensive, the same analysis for A/MS components is still in its infancy [2] and the literature on how do an FMEDA on such circuits is poor [7]. Similarly, pre-silicon verification in the analog domain still implies a substantial amount of manual work and computational effort [9] where EDA tools could contribute to reduce the gap to a complete safety flow.

This paper introduces the Design-For-Safety concept as applied to the traditional EDA for semiconductors applications, and expands on where automation can be useful and, to the best of our knowledge, is still lacking to support the full design/verification/implementation flow. Section II briefly describes the methodology requirements for Automotive applications. Section III summarizes the basic analysis concepts and metrics to capture FS, while Section IV and V analyze how that translates into Design and Verification needs and their respective automation potential. All sections detail the concepts common to digital and A/MS designs and point out where specific differentiation is needed for implementing them and provide guidelines and good practices to face complex A/MS systems. Section VI focuses on the A/MS domain, providing a detailed review of ASIL D Voltage Regulation and a mention to SoC level with a PMIC (Power Management IC) example.

II. Automotive Requirements

This Section describes the automotive methodology requirements in terms of Quality, Reliability and FS and reviews how they are related to each other.

The Failure rate (FR) is the rate at which a component experiences faults. A common representation of the FR of an electronic product during its lifetime is the 'Bathtub curve' (Figure 1). This curve shows three different regions:

- Infant mortality: early life when the product has not reached maturity yet. This is defined as Quality of the product when it exits manufacturing and it is traditionally addressed with Design-For-Test.
- Useful life: after maturity and before wear-out, when the FR is the lowest and mostly a constant value [3]. This is defined

* 978-1-5386-9395-7/19/$31.00 ©2019 IEEE

as the Robustness of the design to several potential effects such as Electro-Static-Discharge, Electro-Migration, random faults due to ionization from particles, etc.

- Wear-out: end of life for the product, when aging and other reliability effects affects the product significantly and the FR raises significantly to become unacceptable.

While, Quality and Reliability are not at all new, automotive poses more stringent requirements than consumer applications:

- The target for Quality is 0 DPPM (Defective Parts Per Million).
- Life expectancy is approximately 15-20 years versus 5 years, stretching the range on which to achieve a good FR
- Temperature scenarios of applications are more extreme, and this worsen reliability effects such as aging, process variation, and electro-migration
- Autonomous driving applications demand high compute power that can only be achieved with advanced technologies, such as 16nm and even 7nm. These are FinFET devices with a significant self-heating that increases reliability challenges.

Reliability guidelines are provided in AEC_Q100 [11], which is a standard for electrical component qualification requirements: Temperature ranges for different grades are included, and automotive is indicated as grade 1 (i.e. [-40°C, +125°C]).

Reliability is defined as the probability that a component will satisfactorily perform its intended function for a prescribed time and under specified conditions [4]: it quantifies the frequency of failures "disregarding the" consequences. The goal of FS, on the other hand, is to ensure the system can move to a safe state despite the presence of a fault: Design-For-Safety amends the architecture with detection circuitry in charge of detecting faults to account for the circuit FR and meet the safety metrics, while reliability effects are modelled with different type of faults (e.g. stuck-at) which are validated with FS verification.

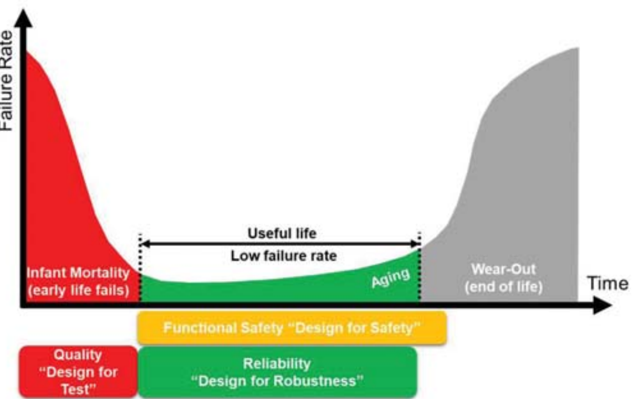

Figure 1. 'Bathtub curve' that represents the Failure Rate vs Time

Quality and Reliability and are well supported and integrated in the traditional EDA flow, therefore this paper focuses on FS: its status and potential for automation. Details are reviewed in Sections III, IV, and V (and anticipated in Figure 2).

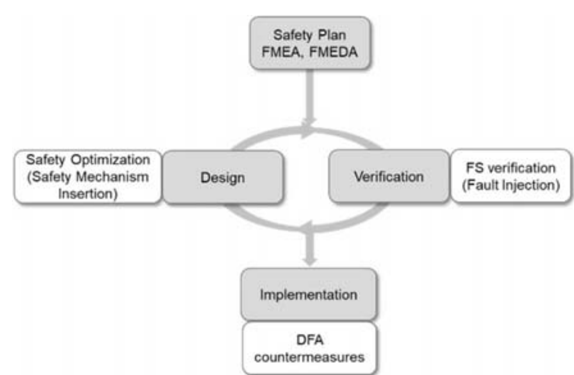

Figure 2. FS linked to the Design/Verification/Implementation flow

III. Analysis

FS analysis is used to evaluate the safety level achieved by the product (e.g., IP, SoC). It comprises quantitative evaluations such as timing analysis, Failure Mode Effect and Diagnostic Analysis (FMEDA), and qualitative assessments such as Dependent Failure Analysis (DFA).

A. FMEDA, Timing Analysis, DFA

FMEDA is a structured approach to define failure modes (FMs), and diagnostic capabilities of a hardware component. Based on the component functionality, the FMEDA hierarchy is structured in parts/subparts/failure modes [5] and basic required inputs include the FR, the Safety mechanism (SM) and its Diagnostic Coverage (DC), i.e. the presence of a safety mechanism and its effectiveness at detecting faults.

			SPFm [%]			99.9
Parts	**Sub-parts**	**Failure mode**	**λperm [FIT]**	**FMD [%]**	**Safety Mechanisms**	**DC [%]**
Low Drop Out Regulator	Regulator Core	Output voltage higher than a predefined high threshold of the prescribed range (i.e. Over voltage)	1.01E-2	21.65	SM1: Under voltage (UV) Monitor SM2: Over-voltage (OV) Monitor	99.9
		Output voltage lower than a predefined low threshold of the prescribed range (i.e. Under voltage)	3.92E-3	8.40		
		Output voltage affected by spikes	1.70E-3	3.64		
		Output voltage oscillation within the prescribed range	1.80E-2	38.58		
		Output voltage fast oscillation outside the prescribed range but with average value within the prescribed range	9.09E-3	9.09E-3		
		Output voltage drift within the prescribed range	2.25E-3	2.25E-3		
		Output voltage drift within the prescribed range	1.60E-3	1.60E-3		

Table 1: FMEDA example for a Low-Drop Out Regulator

The outputs to assess the level of FS readiness are the hardware architectural metrics. The description and formulas that define them are defined in [5]:

- **Single-point fault metric (SPFM) and Latent fault metric (LFM)**: robustness of an item/function to single-point faults and to latent (multiple) faults respectively
- **Probabilistic metric of hardware failures (PMHF)**: provides rationale that the residual risk of a safety goal violation is sufficiently low [6].

Table 1 shows a simplified FMEDA performed for a Low Drop Out Regulator. The SMs are Voltage Monitors providing high coverage for this functionality.

Timing Analysis: The complete evaluation of the SMs involves timing performance: the system must be able to detect faults within the Diagnostic Time Interval (DTI) and transition to a safe state within a specific time, or fault tolerant time interval (FTTI); otherwise, the fault can become a system-level hazard.

The **Dependent Failure Analysis** (**DFA)** also needs to be performed when the system has shared resources: it is a qualitative assessment, also known as analysis of the possible common causes and cascading failures, and it aims to identify the single causes that could bypass a required independence or freedom from interference between given elements and violate a safety goal. Examples of scenarios with potential common cause failures are redundant elements or different functions implemented with identical software or hardware elements.

B. FS Analysis Flow

FS Analysis goes through successive levels of refinement as more information becomes available, as depicted in Figure 3.
A **Qualitative analysis** is performed to identify ways in which the circuit can fail: the design is partitioned into a safety hierarchy of Parts, Sub-Parts and FMs based on the functional description (e.g. a block diagram representation). Several parameters, such as for example the frequency of the FMs, are not evaluated at this stage.

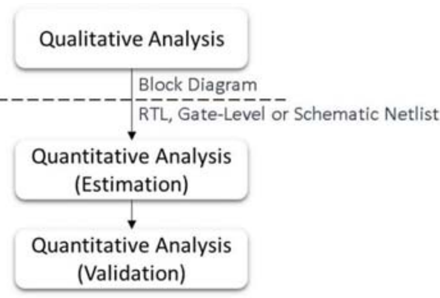

Figure 3: FS Analysis flow

When the design netlist become available (either RTL, gate-level or schematic) a **Quantitative FMEDA** can be performed to predict the Failure Mode Distribution (FMD), i.e. the relative weight of FMs and their probability of failure (FIT). In fact, FMs can now be "connected" to the design components that generate such failures to estimate these values based essentially on area occupation and technology type. Traditionally the connection between the safety hierarchy and the design hierarchy is not formalized or formally captured, hence the estimation work is mostly manual and cumbersome, and usually based on a variety of heuristics. Automation of these calculations is an active area of work in EDA.
When SMs are inserted in the design to reach the desired robustness, their DC can be either **estimated** (based on ISO26262 or expert judgement) or **validated** through FS verification as described in Section V.

C. FMEDA constituents

Figure 4 shows the steps to perform an FMEDA:

1) The component is divided into hierarchical levels (parts, subparts): these are portions of the hardware that can be logically divided
2) For each subpart, FMs (at least one) are defined. FMs describe the way in which an operation potentially fails to deliver the intended function. ISO 26262 [5] provides a list of FMs that can be used for characterizing a semiconductor.
3) The safety hierarchy is linked to the design hierarchy so that FMs are associated to the corresponding part of the design that can trigger them.
4) At this stage the total area of the design can also be evaluated to calculate the total FIT, based on the base failure rate for the technologies deployed.
5) The FM Distribution is evaluated: it expresses the relative weight of a FM with respect to the other FMs of the same Subpart (in other words, the sum of the FMD within a subpart is always 100%). Heuristics are used for this estimation and they can differ significantly between digital and A/MS designs as described Subsection D. Ranking of the FMD and their FIT can be used as criteria to drive the selection of the SMs.
6) SMs are inserted to cover the FMs. Examples of SMs and their classification is reported in Table 2

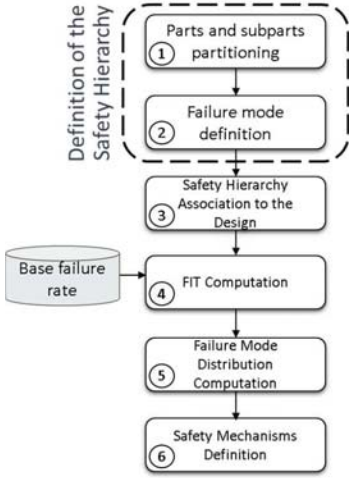

Figure 4: FMEDA constituents

The definition of the safety hierarchy (step 1 and 2 enclosed in the dotted box in Figure 4) is common for the qualitative and quantitative analysis.

D. Functional Safety meets EDA

While Step 1, Step 2 and a large portion of Step 3 are mostly manual and based on the expertise of the Safety Engineer, the other steps can be more easily integrated in an EDA flow with different degrees of automation. Step 3, Step 4 and Step 5 are key to evaluate the total area of the design and estimate its total FIT and the area of each FM. It is evident that with these steps come the high potential for automation. For **digital** circuits, to the best of our knowledge, today the estimation of the area (total and for each FM) can be done semi-automatically aided with some scripting and it can be very detailed or rough based on heuristics. The amount of manual work required in fact depends on how accurate we want the estimate to be and on what other degrees of freedom we are willing to sacrifice to have structures/partitions that can be easily identified and measured. The operation is also inherently complicated by the overlap that can exist in the netlist between different FMs, that need instead to be partitioned properly to correctly account for their contribution to the overall probability of failure. An example of simple heuristic used to estimate the contribution of FMs is to associate one or more output pin to each FM and then estimate the percentage based on the total number of output pins. Clearly, this is a green field where traditional EDA can offer a huge opportunity for automation with its repertoire of netlist partitioning and traversing techniques. For **analog** circuits, the definition and partition of FMs can be more complicated due to the nature of the designs themselves where all the components are more tightly interfering with each other. Often the practical solution to define the FM partitioning is to simulate faults and observe which functions/pins are affected to extrapolate the correlation between the FMs and the faults that trigger the failure. This approach can be limited to design of small/medium size due to the cost of running simulation.

For **mixed-signal**, a mix of the techniques describe above needs to be deployed.

Overall, modern automotive systems are based on very complex circuits and EDA can provide effective tools to automate the processing of the design for safety features.

IV. DESIGN

In this Section, we briefly review how designs can be optimized to meet the safety constraints and how these new requirements can affect the traditional design flow.

A. Design-For-Safety

For safety critical systems, the role of the safety engineer is to devise a FS architecture that includes SMs to meet the required metrics (SPFM, LFM, PMHF). These SMs must be independent from the functions they are protecting to achieve the optimal robustness to failure. It is the FS analysis (FMEDA) that drives the design exploration identifying where to focus the design effort for meeting the constraints. The selection of the best SM for a specific building block or system needs a careful analysis of the tradeoffs between effectiveness for safety metrics, power consumption, area, and timing performance, and even verification time and automation.

To optimize the trade-offs, the benefits and the costs of SMs must be understood in all aspects. To this end, we have created in Table 2 a classification of SMs based on the type and domain of application. SMs based on redundancy replicate the functionality in information, time or space, while diagnostic ones test the functionality itself. While several of the examples direct belong to the digital domain, the classification applies to all types of SMs.

Type	Domain	Description	Family		Examples
Redundant (Exploit the Functionality)	*Information*	Add redundant data to protect the information. Typically works on stored and transmitted information (memories, busses, networks…). More difficult to apply on control logic and self-checking circuits	Parity		Single Parity; Double Parity
			Error-Correcting Code (ECC) Forward-error correction (FEC)	Block-by-block	Hamming; Extended Hamming (SEC-DED) Reed–Solomon
				Convolutional (bit-by-bit)	Viterbi Decoder
	Time	Executing the same operation more than once on the same functional unit, but at different times or sending the same information more than once. Potential high performance overhead	Time Multiplexing		Execute-Retry-Checkpointing-Recovery Configuration Register Test
	Space	Executing the same operation on more than one functional unit at the same time. A voter then selects the final output	Hardware		Dual Core Lock Step (DCLS) Triple-Modular Redundancy (TMR)
			Software		n-version programming Diversity
			Mixed		Simultaneous Multithreading (SMT)
Diagnostic (Test the Functionality)	*Information* *Time* *Space*	Verifies if the intended functionality is working correctly. Potentially corrupts internal status or requiring stopping the online functionality for a given time slot.	Hardware		Logic-BIST; Memory-BIST
			Software		STL; SRAM March Test; Logical monitoring of program sequences
			Mixed		Programmable-BIST; Watchdog

Table 2 : Safety Mechanisms Classification

Also, in complex A/MS system, the diagnostic for the analog portion can potentially resides in the digital domain. In Table 2 we have captured the main characteristics of the most common SMs: their effectiveness (DC) on the safety metrics (both Permanent and Transient) and also their impact on the traditional area/timing performance. In the last column (Notes) we have captured some comments on the "soft" requirements such as automation and verification time. For example, given that verification is such a significant effort, a possible strategy is to select SMs that do not require FS verification even if that entails an additional area penalty. For high ASIL targets, the LFM becomes more difficult to reach and it might require increasing the coverage of the SMs themselves. For example, for the simplest TMR schema where the function is triplicated and then signals are routed through a voter, the LFM is 0 since there is no alarm to report that a fault occurred.

Safety Mechanism	Permanent		Transient	Area/Code Size Overhead	Performance Impact	Notes
	SPFM	LFM				
Single Parity	Low	High (depends on error reporting logic)	Medium	Low	Low	Fault simulation is required to assess the DC on boundary of the logic protected by the parity. Potential to be easily automated.
ECC	High	High (depends on error reporting logic)	High	Medium	Medium	HW verification plan/fault simulation to provide evidence about the correct SM functionality. Potential to be easily automated
DCLS	High	High (depends on error reporting logic)	High	High	Low	HW verification plan/fault simulation to provide evidence about the correct comparator/voter functionality in all the functional/non-functional modes. Potential to be easily automated
TMR	High	Depends on the voter diagnostic	High	High	Low	HW verification plan/fault simulation to provide evidence about the correct SM functionality. Potential to be easily automated
RAM March Test	High (if online within DTI)	High (at startup or online)	High	Low	Medium/High	DC can be very high. Based on ASIL, local fault simulation on the boundary of the memory interface is to assess the correct DC value.
LBIST	High (if online within DTI)	High (at startup or online)	No	Medium	Medium	Requires context-save and restore. Test coverage guaranteed by the quality and number of patterns: Fault simulation used just for signoff activities. Already automated in the digital domain.
STL (Self-test by software)	Medium/Low (depends on #of test segments and if online within DTI)	Medium/Low (depends on #of test segments and if online within DTI)	No	Low/Medium	High/Medium	Fault simulation to be performed. Works best on data-path. Difficult to exploit DC from complex control logic (e.g. fetch and decode) and almost impossible from core performance optimization logic (e.g. branch prediction).

Table 3: Trade-offs of Safety Mechanisms metrics

B. Functional Safety meets EDA

While the SMs categorization and trade-offs applies to both digital and A/MS domains, the potential for automation is clearly much higher for the digital flow. Once the classifications are defined, it is easy to envision an automated flow that encompasses the FMEDA, the design exploration based on it and the insertion of SMs into the design. This optimization step of the Design-For-Safety flow is still a green area for R&D and is mostly targeted for digital, but nonetheless promises to greatly simplify the FS-aware tasks.

V. Verification

This Section covers the basic concepts and terminology of FS Verification and some details on fault modeling. It closes with a discussion on how these concepts can be implemented/automated in the digital and A/MS domains.

A. FS Verification Basics

In the context of ISO26262 part 5 [5] evidence must be provided to support the suitability of the hardware architectural design with respect to detection and control of safety-related random hardware failures. Qualitative information can support the estimation of the DC values, traditionally based on expert judgment, historical information or literature common practices. Therefore, a final FS verification phase might be needed to validate the effectiveness of the same SMs (Figure 3) and confirm whether the design meets its FS requirements: the quantitative nature of the involved metrics (e.g. SPFM) makes direct measure such as fault simulation, one of the most desirable technique to be used especially for higher ASIL and for custom diagnostic.

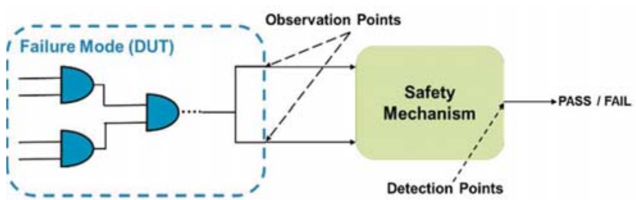

Figure 5: Fault Simulation Environment

Fault simulation is performed in the context of a FM and the SMs targeted to diagnose it: the intended functional behavior is monitored at the **Observation Points**, while the effectiveness of the SMs is checked at the **Detection Points** (see Figure 5) During a fault simulation, the "faulty" circuit response is compared to the one from the 'golden'-fault-free circuit. The

main goal is the **categorization of the faults** injected in the circuit [6], resulting in one of the following classes:

- **Safe**: not propagated to the observation points. Functionality is not affected.
- **Dangerous Detected**: not classified as Safe but observed by one SMs detection points. The detection points can be a physical alarm signal, a memory location etc. where a SM provides evidence of its capability to detect faults [5].
- **Dangerous Undetected:** not classified as Safe, and not observed by any of the SM detection points.

Based on these definitions, it is evident how critical the correct definition of the observation and detection points is: the fault simulator fully relies on these user-defined strobing points to perform its comparison between the 'golden' and 'faulty' simulations, therefore wrong or inaccurate definitions, can lead to misleading results. For a given SM, the DC is defined as the ratio between the dangerous detected and the total dangerous (dangerous detected + dangerous undetected) faults.

Defect Category	Equivalent Defect Abstraction	Comments
DC Short (same layer and in between layers)	R_f	Resistance with R_f value given as a parameter, technology dependent and will become a part of defect coverage report
DC Open (general case)	$Z_f = \frac{R_f}{1 + j\omega R_f C_f}$	Resistance and capacitance given as parameters, technology dependent and will become a part of defect coverage report
DC Open Transistor Gate	$V_{gs} = k * V_{ds}$ $k = \frac{C_{gdo}}{nWLC_{ox}}$	
AC Coupling (same layer and in between layers	C_f	Capacitance C_{fgiven} as a parameter, technology dependent
Resistive Bridge (short)	R_f	Resistance with R_f value given as a parameter, technology dependent and will become a part of defect coverage report
Resistive Bridge (open)	$Z_f = \frac{R_f}{1 + j\omega R_f C_f}$	Resistance and capacitance given as parameters, technology dependent and will become a part of defect coverage report

Table 4: Defect model examples from IEEE P2427 Working Group

Another critical factor to be considered is the **quality of the workload** (expressed in terms of controllability) the fault simulation is executing. The DC is associated to a SM operating on a system that is supposed to execute its intended functionality (as a side note, this is the major difference compared to the test coverage coming from DFT/ATPG). In this context, if the fault simulation is not correctly stimulating the system functionality, the end results in terms of number of safe, dangerous detected/undetected faults will be misleading: good coverage of the functional verification environment is a prerequisite of the FS verification task.

B. Fault Modeling

Digital: For digital circuits, relatively simple fault models stuck-at-0 (ST0) and stuck-at-1 (ST1) have proved sufficent for analysing fault coverage. The analysis is further simplified by the fact that for primitive functions (gate-level), the faults can be injected only at the borders (input and output pins).

Analog and Mixed/Signal: For analog designs, defining faults has proven to be much more of a challenge. Certainly, ST0 and ST1 are not sufficient, but what is sufficient is still being defined. Are catastrophic faults (i.e., shorts and opens) enough or are parametric faults also needed? Is it possible to define faults for primitive technology library components (e.g. current mirrors, differential pairs) as for the digital domain? The lack of analog fault models can be a challenge when trying to define a methodology for fault simulation test coverage. Standard analog faults are identified from analyzing technology process reliability data: new analog defect models are proposed in IEEE P2427 that is still being worked upon and reported in Table 4. Aging and reliability simulation data are used to determine weighted likelihood of component faults, including specific MOSFET transistor SPICE parameter(s) variation that may exceed maximum allowed ±6σ variation during the expected device lifetime usage.
Functional Safety meets EDA

FMEDA drives verification as much as design optimization and the fault injection campaigns to validate the DC of SMs are usually executed on a per FM basis. Once the FMs (and their Observation Points) and the SMs (and their Detection Points) have been associated to parts of the design, it is straightforward to infer automatically where the faults should be injected and observed/detected for the fault categorization. Though the concept applies to both digital and A/MS, the top-down implementation in serial steps as described above applies mostly to the digital domain and A/MS still requires a more custom approach. In fact, the root cause of FM in analog are located in different sub-modules/sub-blocks and multiple components within these can contribute with their faults to the defined FM (e.g. "output voltage regulation too low or under-shooting" for the Voltage Regulator VREG example). An analog fault simulation campaign is performed first to define which faults are contributing to defined FM and at the same time is used to validate the DC value of the SMs. This data combined with the physical size of the corresponding analog component is used to accurately calculate the FM distribution value. Extension of the traditional verification solution to FS verification is in high demand and the EDA industry is actively working on it. For example, the fault injection campaign can be managed and optimized using techniques such as fault list reduction (based on structural analysis) and test optimization (dropping, merging). A/MS fault injection can be a very expensive task and therefore it is critical to minimize the fault list (grouping) [8] to be simulated without missing potentially safety-critical faults. Further automation can be achieved by classification of the SMs so that the correct type of fault simulation setup can be generated accordingly. In fact, SMs can operate **periodically** (e.g. BIST, STL etc.) during each DTI or

continuously (e.g. ECC). They can work in the same functional path of the circuit they protect (**active**) or just monitor the circuits functionality to detect an anomalous behavior (**passive**). The classification between 'active' or 'passive' SMs has an impact on the safety metrics, since the former can contribute both to SPFM and LFM, while the latter only to LFM. Similarly, the classification of periodic and continuous SMs implies a different type of simulation setup and measurement for fault categorization. Examples in Section VI will further elaborate on this concept.

A last word is to be spent on **scalability**: fault injection is an expensive task, and, in the analog domain, it is realistic only for relatively small designs and it is not practical at system or SoC level. Often, the problem is mitigated by the fact that digital and analog fault simulations can be separated and then results combined for any given FM. A true A/MS simulation, potentially based on behavioral modeling (e.g. Verilog-A; VHDL-AMS) is still in the discussions for reasons that are both technical and related to development of a standard. The concept is very similar to the fact that digital fault injection of permanent faults is considered accurate only at the gate-level: more research and widely-recognized results would be needed to accept a higher-level, potentially less accurate, methodology. Part of the automotive paradigm, is in fact, confidence achieved with use, which is still to be developed in this specific topic.

VI. Mixed-Signal Application Examples

In this Section, we present two examples of ASIL D Voltage Regulation circuit implementation with data processing and decision making in the digital core and review how the different requirements for Design-For-Safety apply to these A/MS circuits. In both cases, the SM are themselves covered by diagnostic, which makes these implementations suitable for ASIL D targets. In example #1 (Figure 6), the SMs for the Voltage Regulation circuit is the Voltage Monitoring circuit: they are independent of each other, with the outputs of the voltage monitoring sampled by the digital core. In addition, there is a dedicated periodic Analog Built-In-Self-Test (BIST) circuit for voltage monitoring diagnostics. In example #2 (Figure 7), the Voltage Regulation circuit and the Voltage Monitoring circuit are again independent of each other, with the voltage monitoring outputs sampled by digital core. Here, in addition, a redundant voltage monitor (VMON-2) is introduced as diagnostic to VMON-1, instead of the Analog BIST. VMON-2 is compared against the VMON-1 in the digital core as a plausibility check. The Voltage Monitor SM includes an analog function (voltage comparator circuit) and a digital one (voltage comparator de-glitch function, status bit and decision-making logic). The SPFM and LFM metrics are the sum of the respective analog and digital function DC values.

Below are some considerations regarding FS verification of the Design-For-Safety flow which apply to both examples:

The analog and digital fault injection simulations can be done separately. In fact, there is no test case which requires fault injection in a digital domain and its propagation into the analog domains, and vice versa.

- Faults at the Analog–Digital interfaces are modeled as input port faults of the analog and digital modules respectively.
- For the SPFM DC validation, UVM (Universal Verification Methodology)/SV (System Verilog)/VAMS (Verilog-AMS) assertions are used on the VREG output to confirm whether the voltage regulation goes out of range with the injected fault, while asserts on the VMON outputs are to check whether the dangerous fault is detected or not.
- The statistics on which component and which of its faults contribute to the observed analog FM are combined with the area information to accurately determine the FMD value.

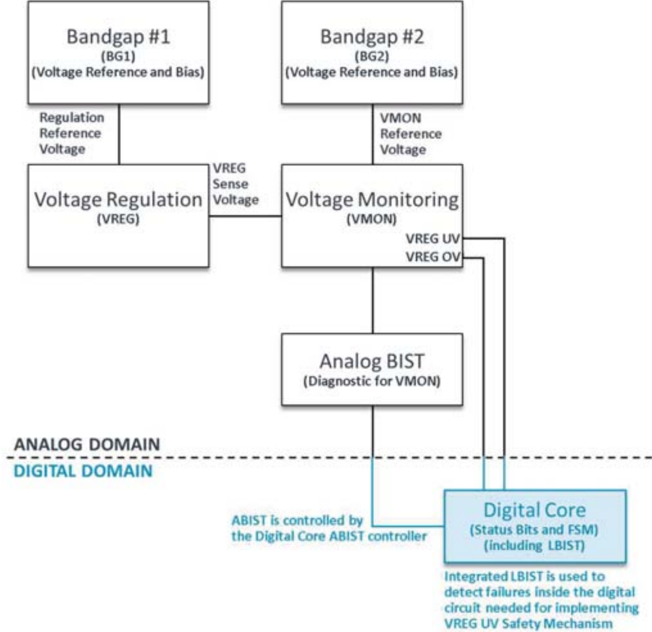

Figure 6: Voltage Regulation and Monitoring example #1

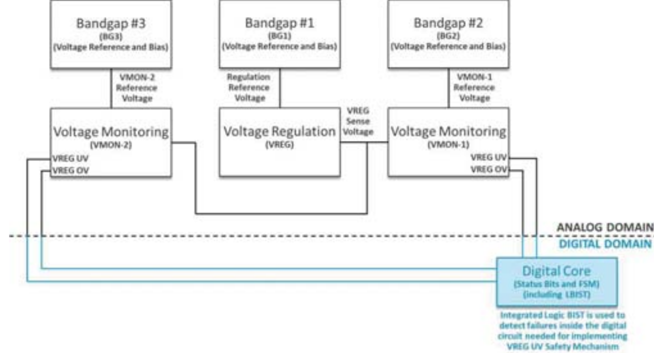

Figure 7: Voltage Regulation and Monitoring example #2

These two implementations highlight an interesting difference between the fault injection setup for periodic versus continuous SM, as mentioned in Section 0. The observation is about the LFM metric validation, which verifies the DC of the SM itself.

In the example #2, the LFM metric can be measured in a very similar fashion to the SPFM metric detailed above: faults are injected in VMON1 and assertions are used to check whether they are detected by VMON2.
In the case of example #1, instead, a two-step fault injection simulation setup is required to handle the Analog BIST:

- Step 1: Fault injection is run setting the BIST in functional mode to identify all faults that lead to the VMON SM not detecting an output voltage out of regulation, or falsely detecting output voltage out of regulation. This generates the list of faults affecting the VMON functionality.
- Step 2: Fault injection is run using the fault list generated in Step 1 (i.e. only the dangerous faults) and setting the BIST in scan mode to confirm whether the fault is detected or not.

These examples show how EDA FS flow automation has a significant opportunity to relieve much of the manual work that is still in place.

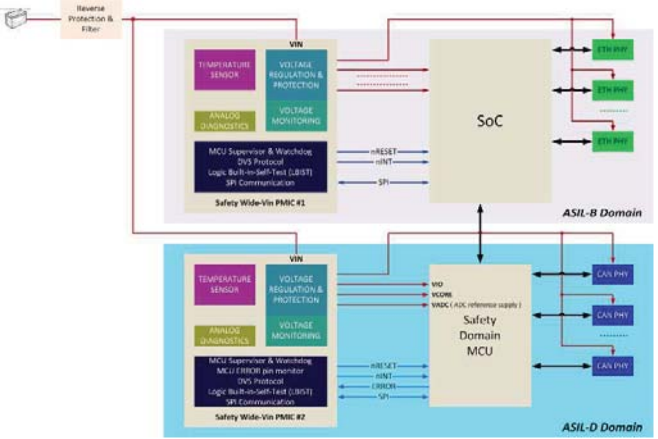

Figure 8: Examples of PMIC (Power Management IC) system

The last examples shown in Figure 8 put the Voltage Regulation in context of the SoC and illustrate different power supply functional requirements for PMIC #1 and PMIC #2 and how that translates into different FMs (and, consequently, into different analog components being the FM root cause). The new generation of SoCs do not only require accurate voltage supply (DC specification), but also very fast load step response and low voltage ripple (AC specification). Random failures of PMIC #1 that are leading to violation of the voltage regulation AC are considered safety critical, while the same failures may not need to be considered for PMIC #2 since the supply rails for the Safety Domain MCUs are in general less sensitive (with exception for ADC reference supply and system required ADC measurement accuracy).

Several safety features apply to both PMICs:

- Supervision and watchdog monitoring functions are needed and must be analyzed against the digital core FMs. Dynamic Voltage Scaling (DVS) may be required for the SoC to manage power consumption (and thermal performance): for DVS analysis not only the analog FMs related to regulated supply rails for SoC have to be considered, but also the SPI (Serial Peripheral Interface) communication protocol failures related to the digital core.
- Voltage monitoring functions contain an analog function (2nd reference voltage source, sense feedback resistive network and voltage comparators) and a digital function (analog comparator digital de-glitch filter, SPI memory mapped status bits and decision-making logic based on the detected voltage regulation failure).
- Die temperature sensors supplement integrated voltage regulation protection against over-load conditions.

Fault injection simulation results can validate the qualitative PMIC safety architecture analysis to prove independency of the voltage regulation circuits from the voltage monitoring circuits, as well as independency of the temperature sensors from the voltage regulator over-load (or current limit) functions.

VII. Conclusions

In this paper we have reviewed the methodology requirements for safety-critical applications, such as autonomous driving. The discussion has focused on the Design-For-Safety paradigm which is not yet fully finalized and supported in EDA tools. Some of the concepts where illustrated on the Voltage Regulation and Power Management examples, in the A/MS domain. There is a great deal of activity and clearly a significant potential of contribution to be made in this space.

References

[1] Meany, Tom. "Functional safety and Industrie 4.0." Signals and Systems Conference (ISSC), 2017 28th Irish, IEEE, 2017.

[2] Fraccaroli, Enrico, et al. "Fault analysis in analog circuits through language manipulation and abstraction.", Specification and Design Languages (FDL), 2017 Forum on. IEEE, 2017.

[3] R. F. Stapelberg, "Handbook of reliability, availability, maintainability and safety in engineering design", Springer, 2009.

[4] R. Mariani, "The impact of functional safety standards in the design and test of reliable and available integrated circuits", 17th IEEE European Test Symposium (ETS), 2012.

[5] ISO 26262:2011 Road vehicles - Functional Safety.

[6] A. Nardi, A. Armato. "Functional Safety Methodologies for Automotive Applications", 2017 International Conference on Computer-Aided Design (ICCAD).

[7] E. Fraccaroli, F. Stefanni, F. Fummi, M. Zwolinski, "Fault analysis in analog circuits through language manipulation and abstraction", 2017 Forum on Specification and Design Languages (FDL).

[8] Oezlem Karaca, et. al., "Fault grouping for fault injection based simulation of AMS circuits in the context of functional safety", 2016 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD).

[9] S. Simon, et al., "Safety-oriented mixed-signal verification of automotive power devices in a UVM environment", 2016 13th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD).

[10] SAE J3016: "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems"

[11] AEC-Q100-Rev-H: "Failure Mechanism Based Stress Test Qualification for Integrated Circuits", Automotive Electronics Council, Component Technical Committee, released Sept. 11, 2014.