# FuSa in a Nutshell - Fail operational systems and active suspensions

## Safety Lifecycle use case

## About this document

### Scope and purpose

Meeting functional safety (FuSa) standards in automotive, industrial, and other domains are often challenging. This document provides a first set of guidelines for users who are unfamiliar with using Infineon automotive MCUs in a functional safety scope. This application note is part of a series of documents named “FuSa in a nutshell”, as listed in [2].

### Intended audience

Intended audience are functional safety engineers and designers, including AURIX™ MCU users who are new to FuSa, interested in hardware-based implementation, and seeking safety details beyond the MCU user manual.

### Structure of the document

This document illustrates the use of Infineon automotive MCUs within the lifecycle of a fail-operational system, supported by an illustrative example using AURIX™ TC4xx family.

### Disclaimer

**Application use cases, if any, are for training purposes only and are not to be taken as a blueprint for productive development.**

### Keypoints

- Introduction to fail-operational system development cycle
- Use Active Suspension Control (ASC) as an illustrative example to link functional safety concepts to a real-world application scenario

## About this product family

### Product family

The AURIX™ MCU family is designed to meet the high-performance, safety, and security requirements of modern automotive and industrial applications. Targeted for ISO 26262 ASIL-D compliance as a Safety Element out of Context (SEooC), AURIX™ devices provide functional safety support for safety-critical systems.

### Target applications

- Steer by wire and brake by wire
- Fail operational power supply and distribution
- Fail operational active suspensions

# Table of contents

**About this document** ..... **1**
**About this product family** ..... **1**
**Table of contents** ..... **2**
**1 Safety lifecycle activities in fail-operational systems** ..... **3**
**2 Concept phase** ..... **4**
2.1 Item definition ..... 4
2.2 Preliminary vehicle architecture of Active Suspension Control ..... 4
2.3 Hazard analysis ..... 5
2.4 Safety goal and ASIL ..... 7
2.5 Functional safety concept ..... 8
**3 Product development at system level** ..... **12**
3.1 Dependent failure and common cause ..... 12
3.2 System Faults, monitors, and reactions ..... 14
3.3 Allocation of safety requirements ..... 15
**4 Hints for AURIX™ TC4xx family in Fail-Operational context** ..... **17**
**5 Safety analysis techniques for fail-operational system** ..... **19**
**6 Related resources** ..... **20**
**References** ..... **21**
**Glossary** ..... **22**
**Revision history** ..... **24**
**Disclaimer** ..... **25**

# 1 Safety lifecycle activities in fail-operational systems

The safety lifecycle refers to the set of additional activities conducted in conjunction with the standard development process for safety-related systems. For the purpose of this application note, the safety lifecycle activities considered is not complete, as the production and subsequent phases are outside its scope. The target of this document is to guide designers on how to approach the safety development of fail-operational systems in compliance with ISO 26262 standards.

Key informations needed for these activities includes:

- Indentifying the reason **why fail operational capability** is required
- Defining **acceptable Emergency operations** when a fault occurs
- Analyzing **dependencies** between intended function and its monitor/fault reaction

In Figure 1 , the safety lifecycle activities are conceptually illustrated to explain the commonality with the classical V-cycle method and aspects specific to fail-operational systems, such as architecture and emergency operation strategy.

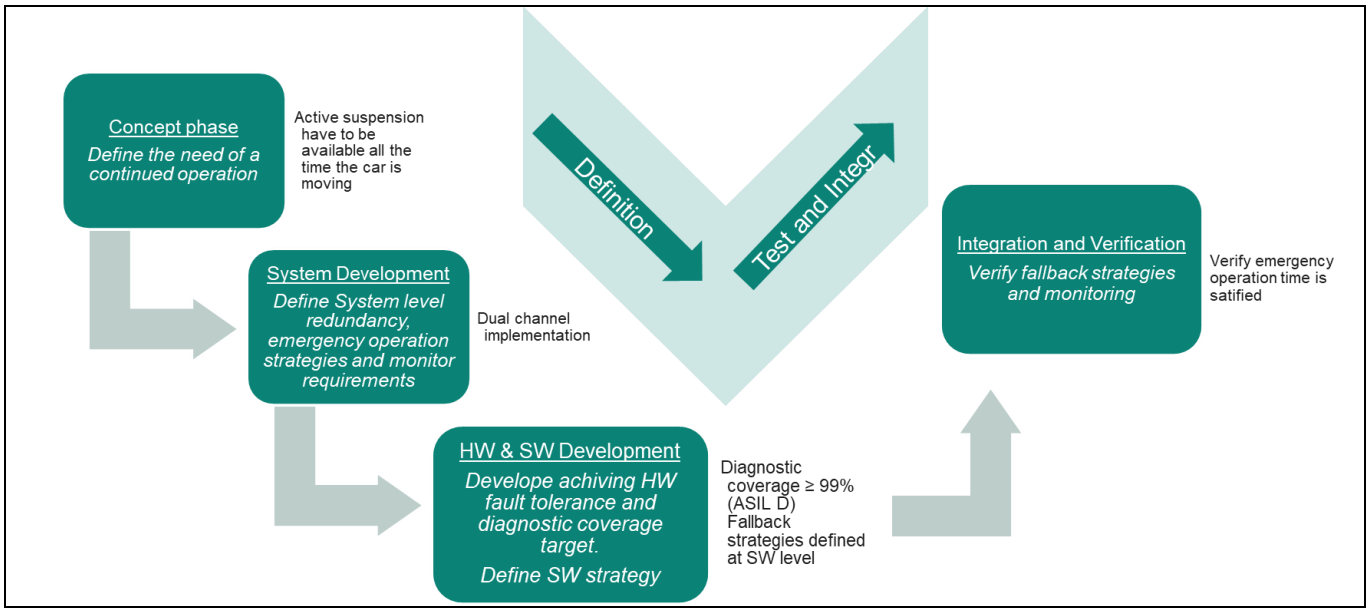



**Figure 1 Safety Lifecycle activities**

To support the theoretical presentation of the safety lifecycle in fail-operational system, this document uses an Active Suspension Control (ASC) system as an illustrative example.

***Attention:*** ***The ASC system example is provided purely for illustration. It is not exhaustive and does not cover the full set of possible architectures, hazards, or design choice. The example serves a consistent reference point that is revisited at the end of each subchapter, to help the reader to connect general concepts to a tangible use case.***

# 2 Concept phase

The concept phase in ISO 26262 establishes the foundations for safety by:

- Defining the item
- Defining the item context, alias the vehicle architecture
- Hazards analysis, where malfunctioning behavior are evaluated
- Stating safety goals and their related ASIL
- Deriving functional safety concept starting from safety goals

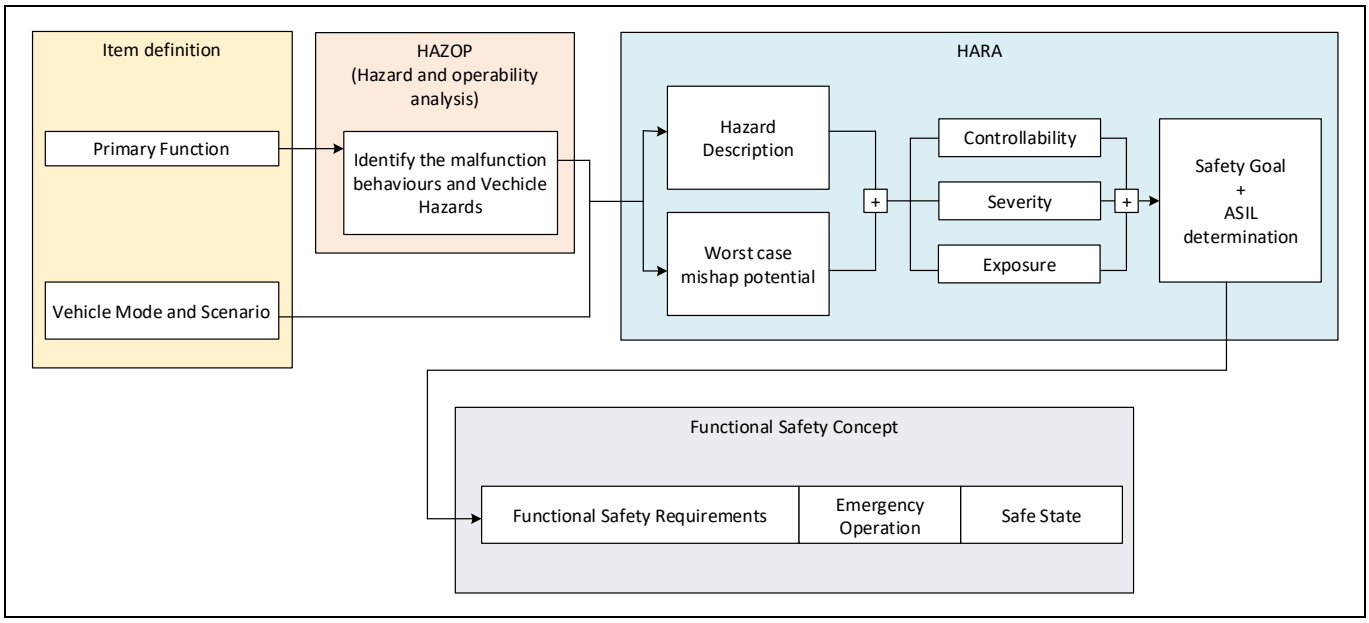



**Figure 2 Concept phase steps**

During the concept phase, the need for fail-operational capability is first identified: certain functions cannot be simply shut off when faults occur, as doing so would itself create an unreasonable risk.

## 2.1 Item definition

The item definition describes the systems under analysis, including its scope, boundaries, environment, and interfaces. This step ensures a clear understanding of what is included in the safety process. For fail-operational systems, item definition is particularly important because the ability to maintain a function after faults depends on how the system interacts with other vehicle domains (e.g., power supply, communication buses).
As an example, let us consider as active suspension control system. Its environment includes the vehicle power supply (e.g.,48 V), the chassis domain controller, and the mechanical suspension elements. The scope excludes non-safety-critical comfort functions, such as adaptive damping. Clear boundaries ensure that fail-operational requirements are properly attributed to the suspension system itself, and not to external vehicle functions.

## 2.2 Preliminary vehicle architecture of Active Suspension Control

The preliminary block diagram of Active Suspension Control at vehicle level is shown in Figure 3. From this schematic, the interaction requirements will be derived.

The following is a list of the systems that will be analyzed and considered for this application:

- **Electronic stability control (ESC):** Responsible of the dynamic of the vehicle
- **ASC (FL, FR, RL, RR):** Active suspension control for each corner
- **Power Supply Sources:** Power supply for active suspension
- **Communication Lines:** Communication paths between components

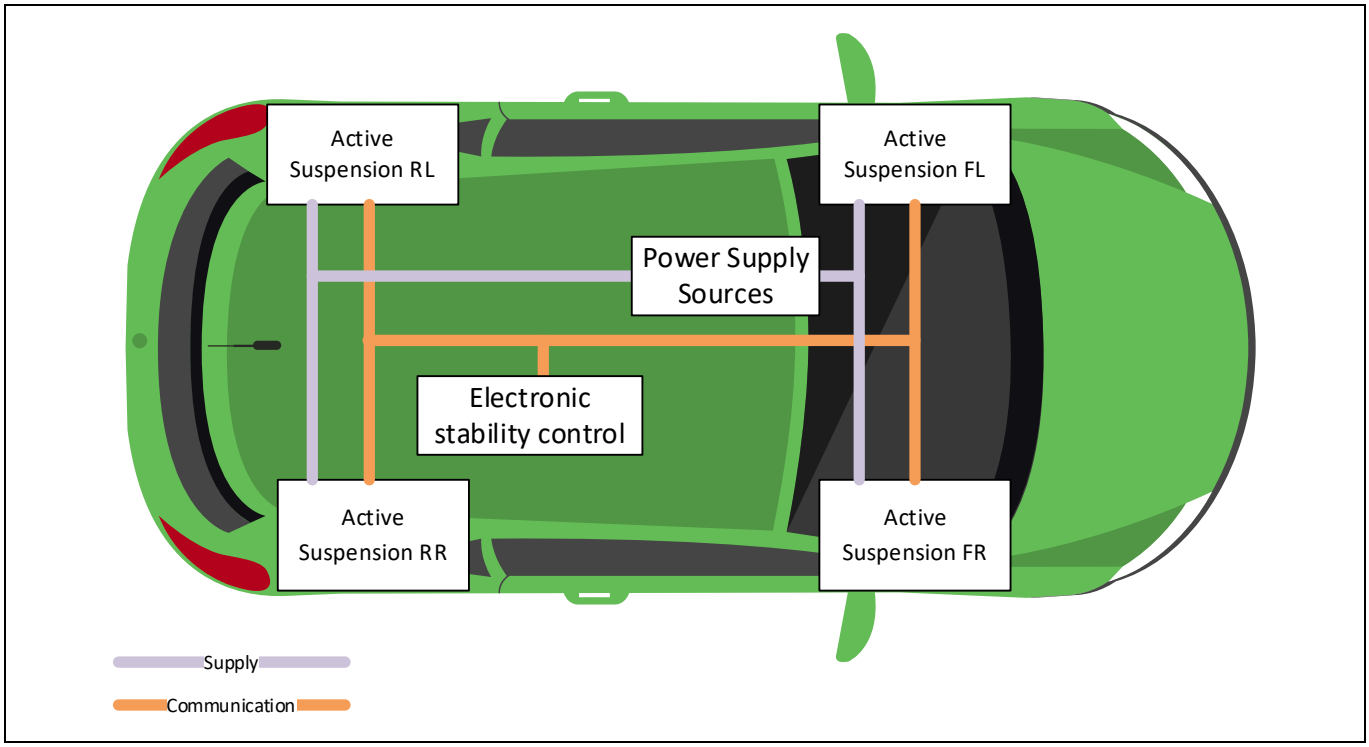



**Figure 3 Active Suspension Control (ASC) block diagram at vehicle level**

## 2.3 Hazard analysis

This section is not intended to provide a detailed explanation of how to conduct a Hazard Analysis and Risk Assessment (HARA). Instead, its purpose is to clarify why the need for fail-operational systems can emerge.

A natural question arises: **why were fail-safe systems previously sufficient for most components, but are no longer adequate today?**

This “shift” is driven by the evolution of modern vehicles, which are increasingly designed toward reducing human intervention. In such systems, the driver adopts an entirely passive role during operation, as is the case of autonomous vehicles described in Figure 4.

## Safety Lifecycle use case

### Concept phase

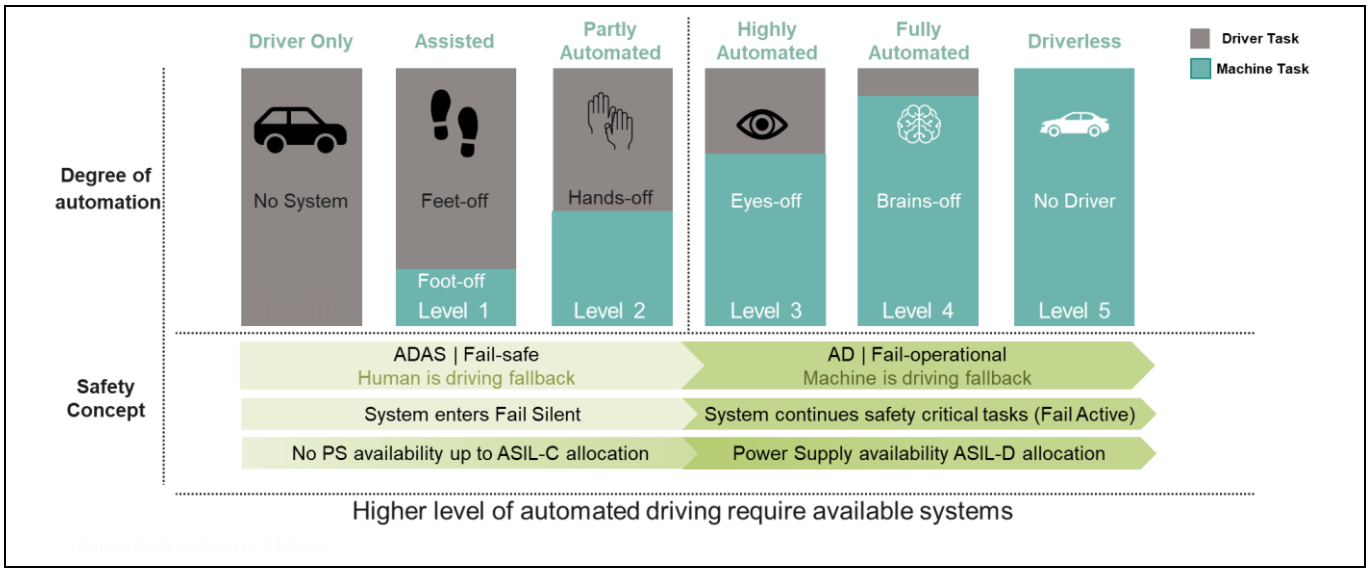



**Figure 4 SAE levels for automated driving**

A preliminary hazard analysis can be performed to identify situations requiring continued operation. The identification process for possible hazards can started by examining the categories listed in Table 1.

**Table 1 Possible types of hazards**

| Hazards | Description |
|---|---|
| No function | Function not provided when needed |
| Partial/Intermittent/Over Function/Degraded Over Time | Incorrect function when needed |
| Unintended Function | Function provided when not needed |

Let us provide this preliminary hazard analysis of the ***active suspension functions***. A possible hazard for this scenario could result from **unintended wheel height variation** during a hairpin turn or obstacle avoidance. Since the response has to be fast and the reliability is critical, a possible solution can be a fault-tolerant system.

Here, the controllability by the driver is not assumed, as we are considering an autonomous driving (AD) vehicle, and the described malfunction could induce yaw or roll moments in the vehicle. This scenario could be compliant with a safety goal rated ASIL C (see Table 2).

**Table 2 Preliminary hazard analysis**

| Primary function | Vehicle hazards | Worst mishap potential | Controllability | Severity | Exposure | ASIL |
|---|---|---|---|---|---|---|
| Active suspension | Loss of vehicle stability (yaw or roll moments in the vehicle) | Unintended wheel height variation in hairpin turn | C3: Sudden asymmetric ride height in a tight turn creates immediate yaw/roll; the AD can't realistically counter-steer or brake in time | S3: Loss of stability can lead to rollover or complete loss of control → life-threatening injuries/fatalities | E3: Tight hairpins at speeds where active height changes are relevant are present on most trips | ASIL C |

By analyzing the system, we can also assume that the **power supply is fault-tolerant,** as the absence of a power supply can directly affect the system's ability to maintain a safe suspension position. Ensuring the power supply would be the responsibility of another system, such as the battery management system (BMS).

A similar evaluation can be performed for the **steer-by-wire system or brake by wire system**, where loss of functionality can also have catastrophic consequences if not properly managed.

In summary, if a human or passive fallback is not possible, the item must be classified as fail-operational. This means that the system is designed to continue operating safely, even in the presence of a fault, until a safe condition is reached.

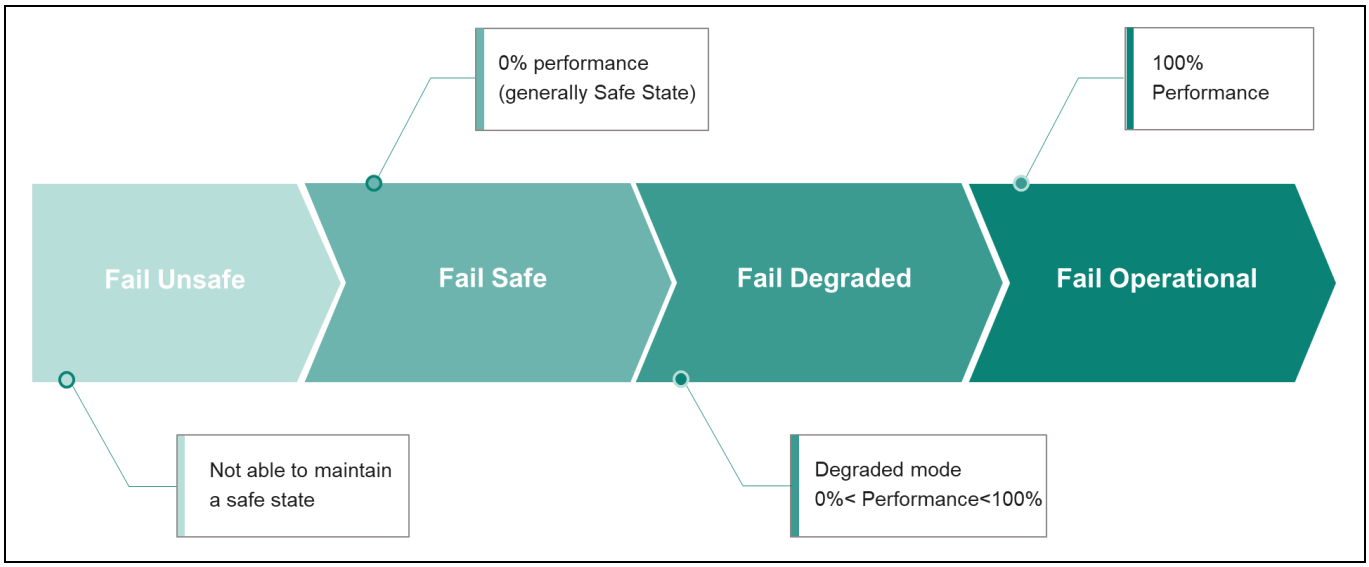



**Figure 5 Logic to decide when to consider a system “fail operational”**

## 2.4 Safety goal and ASIL

Based on the identified hazards, safety goals must be formulated to define what the system shall prevent. These safety goals should be specific, measurable, and achievable.

**Table 3 Safety Goal definition**

| ID | Description | SG ASIL | Operating mode |
|---|---|---|---|
| SG1 | Prevent loss of vehicle stability due to suspension failure | ASIL C | Normal (running and no fault) |

For example, a possible vehicle safety goal could be: **Prevent loss of vehicle stability due to suspension failure**. This safety goal clearly defines what the system has to prevent in order to ensure safe operation.

Each safety goal inherits an Automotive Safety Integrity Level (ASIL) from the Hazard Risk Classification (based on Exposure, Controllability, and Severity). This ASIL rating determines the level of safety required for the system.

## 2.5 Functional safety concept

Starting from the safety goals, it is possible to develop the system's functional safety concept required to provide a **fault-tolerant active suspensions system**.

The following are the functional safety actions chosen for this use case:

- **Fault detection (monitor):** Ability to detect a possible fault in every situation
- **Emergency operation (reaction to single fault):** Designing the system to continue operating safely, with or without performance degradation, even in the presence of a fault
- **Transition to safe state (system reaction to minimize the risk of harm):** Ensuring that the system can transition to a safe state if one or multiple faults are present

These strategies will help to ensure that the safety goals are met, and that the system is designed to be fault-tolerant and safe.

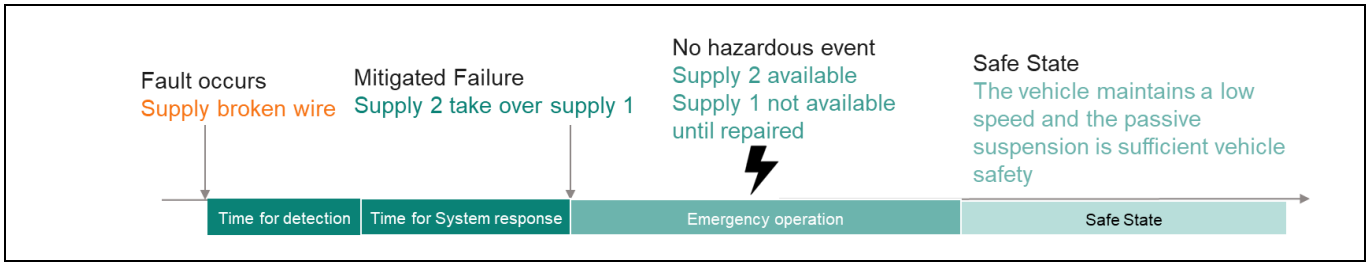



**Figure 6 Time sequence in case of fault in a fail-operational system ( power supply example)**

In Figure 7, the strategy used for this use case, called "cold" redundancy, is shown. In particular, operating modes are:

- **Power ON:** Start-up test ensures the correct operation of the channels
- **Normal Behavior:** Primary channel works in full operational mode
- **Emergency Operation:** Switch over to second channel in case of first fault on the primary channel
- **Safe State:** Proper safe state is provided depending on the application
- **Power OFF:** Actuators are disabled and faults are recorded because they will be checked in next start up

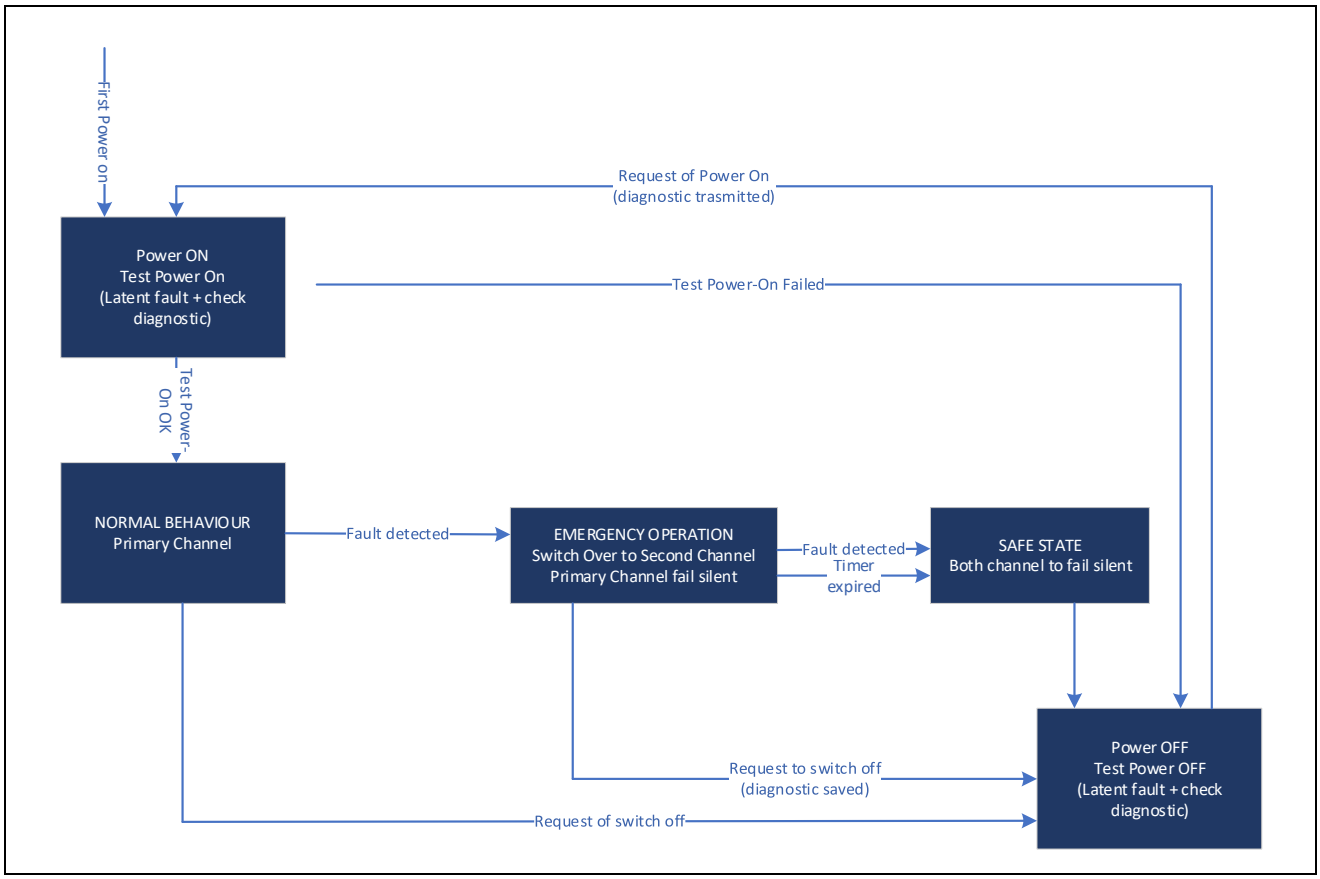



**Figure 7 Operating Mode**

In the scope of this document (Active Suspension Control use case), repair of the first channel is not considered, for simplification reason.

Fault types to tolerate remaining operative:

- Power Supply fault
- ESC fault
- Communication loss (CAN/Ethernet)
- ASC fault

The component shall be developed in accordance with the principles showed in Figure 8, where independence between monitor and intended function, as well as command 1 and command 2, is the key concept.

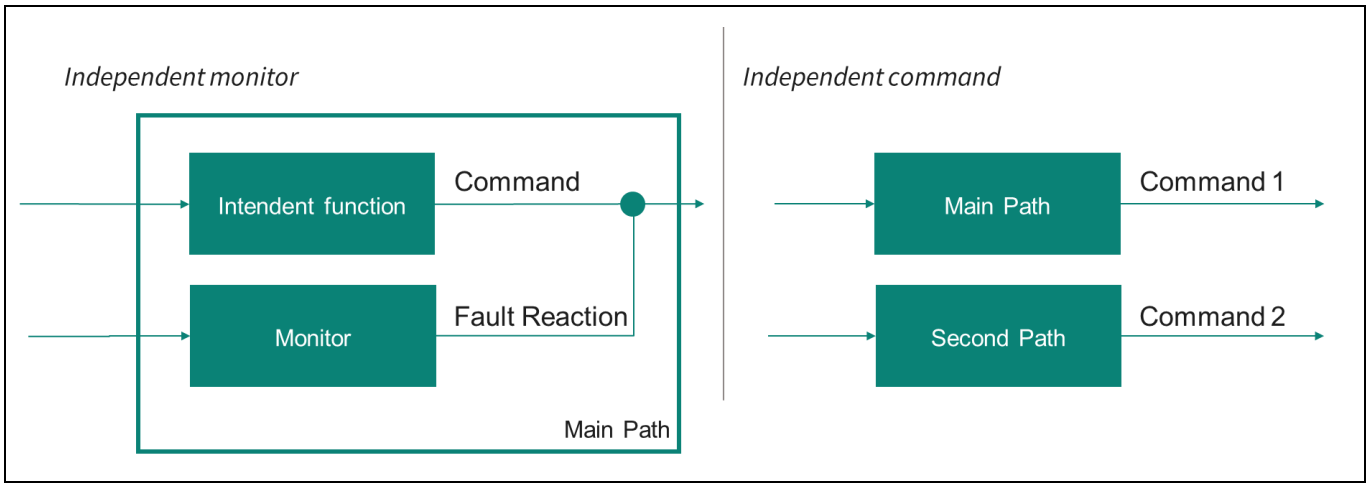



**Figure 8 Independent monitor and path**

Starting from the safety goal, functional-safety requirements are derived and associated with one of the systems analyzed.

According to ISO 26262, for each functional safety requirement a defined fault reaction – known as the emergency operation requirement– is proposed. Additionally, for each functional safety requirement the safe state, that has to be reached before the emergency operation tolerance time interval (EOTTI), is proposed. An easy example of this sequence, (FSR+EO+SS) can be seen in Figure 9.

**Table 4 Safety goal and functional safety requirements net**

| SG_ID | System | FSR_ID | Functional safety requirements | Emergency operation requirements (1 fault) | Safe state (reached before the EOTTI) |
|---|---|---|---|---|---|
| SG1 | ASC | FSR_ASC1 | Provide safe height of active suspension to prevent malfunction that can affect vehicle stability | Provide alternative independent safe high of active suspension | Passive suspension + Communication off (fault managed by ESC) |
| SG1 | Supply | FSR_SS1 | Provide safe supply to prevent propagation of wrong supply | Provide alternative independent safe supply | Supply disabled + Managed at vehicle level not to reach an unreasonable level of risk + warn driver |
| SG1 | ESC | FSR_ESC1 | Provide safe communication to prevent unintended suspension actuation | Provide alternative independent safe communication | Communication disabled (active suspension in passive mode) + Managed at vehicle level not to reach an unreasonable level of risk + warn driver |

| SG_ID | System | FSR_ID | Functional safety requirements | Emergency operation requirements (1 fault) | Safe state (reached before the EOTTI) |
|---|---|---|---|---|---|
| SG1 | ESC | FSR_ESC2 | ESC has to ensure that the faulty channel have to be available again before a time threshold (Teotti) to prevent malfunction that can affect vehicle stability | Provide alternative independent system that has to ensure that the faulty channel have to be available again before a time threshold | Managed at vehicle level not to reach an unreasonable level of risk + warn driver |
| SG1 | ESC | FSR_ESC3 | ESC has to ensure vehicle stability in case of passive suspension (active suspension disabled) to prevent malfunction that can affect vehicle stability | Not Applicable | Manage at vehicle level not to reach an unreasonable level of risk + warn driver |

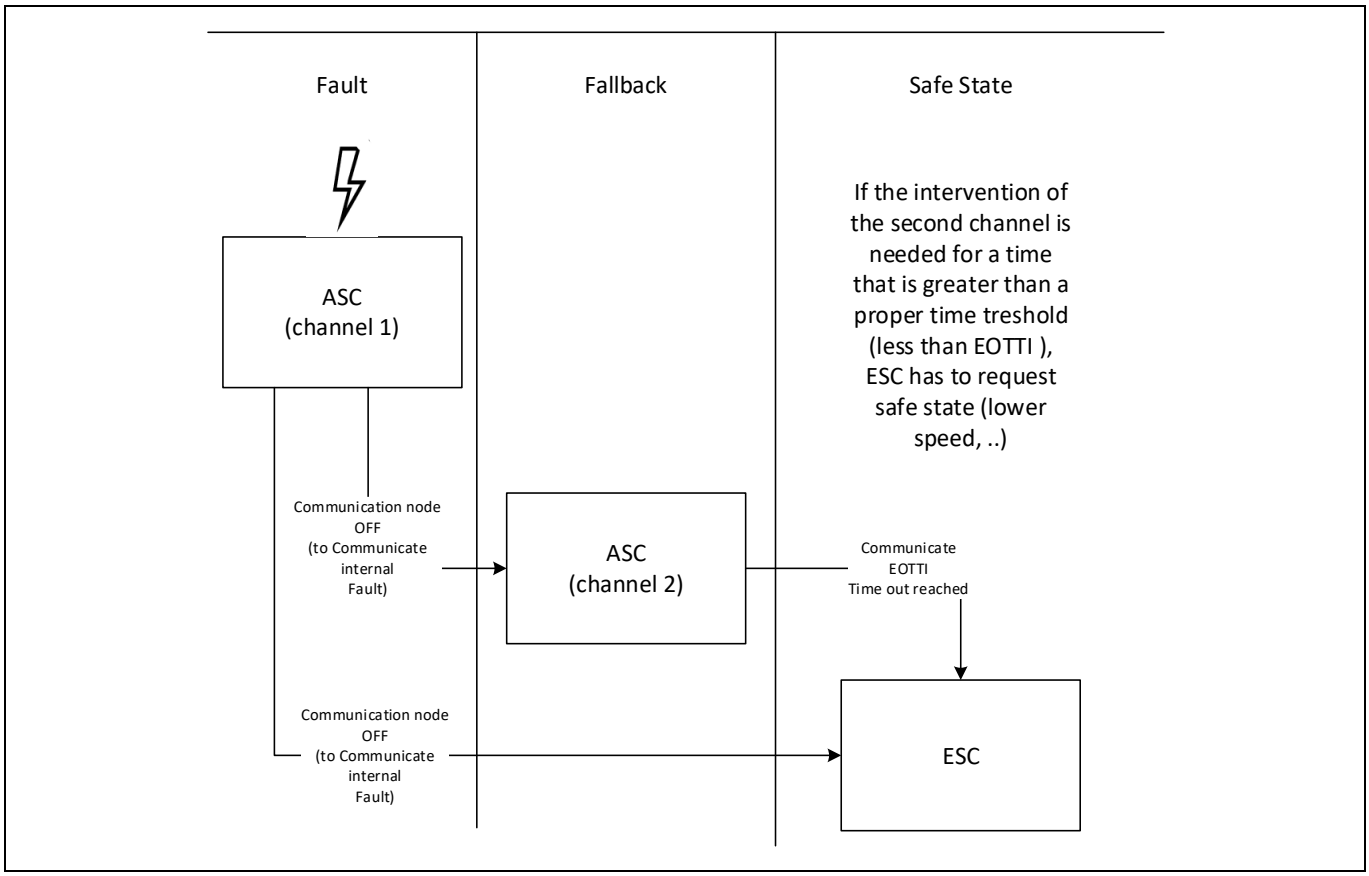



**Figure 9 Example of functional safety requirement behavior (emergency operation + safe state)**

# 3 Product development at system level

This section covers the product development at the system level, where the functional safety concept is refined into a concrete system architecture design and its technical safety requirements.

The goal of this phase is to design a safety architecture that can meet the safety goals and allocate safety requirements. To ensure this, safety analysis is performed to demonstrate the system's capability to meet the functional safety requirements required for the fail-operational use case.

The system we consider for this analysis consists of four electromechanical actuators, a dedicated Active Suspension Control ECU, suspension height sensors, and communication links with the vehicle network.

During this phase, the system-level redundancies, degraded operation strategies, and monitoring requirements will be defined. In general, a fault-tolerant system would require redundant components, such as:

- **Power supply:** Continued operation in case of a power failure
- **Sensing:** Multiple sources of data for fault detection and mitigation
- **MCU Logic:** Provide the intended function / electronic control of Active Suspension Control
- **External monitor (watchdog)**
- **Actuation:** Multiple means of controlling the system
- **Communication:** Continued communication between ECUs

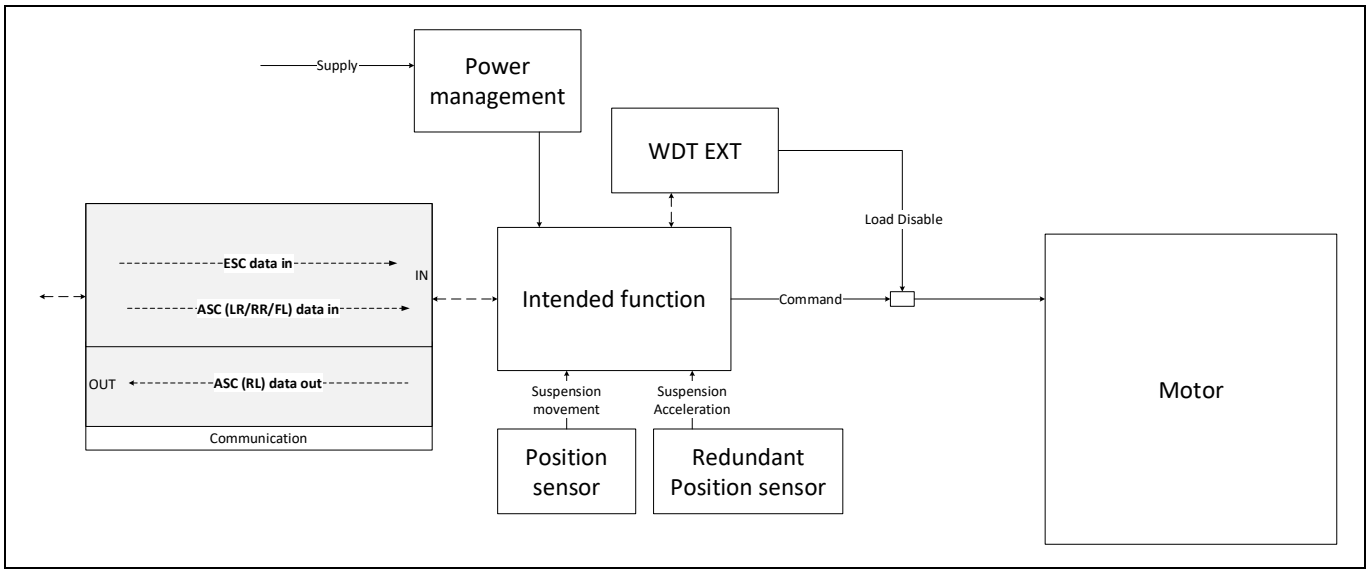



**Figure 10 ASC Single channel fail safe: intended function + independent monitor**

## 3.1 Dependent failure and common cause

A fail-operational architecture is effective only if the channels are independent. Therefore, verifying the safety analysis is crucial to confirm the independence of these redundant channels.

Avoiding dependencies can be done through the following activities:

- HW separation and SW partitioning shall be evaluated
- SW redundancy using diverse algorithms with cross-check of the results

Activities to validate the design could be (not exhaustive list):

- Fault Tree Analysis (FTA)
- System Failure Mode and Effects Analysis (System FMEA)
- Perform a Dependent Failure Analysis

*Note:* *As fail-operational does not mean that the system is operational under multiple simultaneous faults, only one fault is tolerated. The occurrence of second fault moves the system to a safe state. Each requirement is associated with a safe state and is verifiable.*

## 3.2 System Faults, monitors, and reactions

A fail-operational design requires robust fault detection for each channel and a coordinated reaction strategy.

In Table 5, the Technical Safety Requirements (TSRs) are defined starting from the previously defined functional safety requirements.

**Table 5 Functional safety requirement and analysis of system monitor and reaction**

| FSR | Element | Element fault | TSR | |
|---|---|---|---|---|
| | | | System Monitor | System reaction |
| Provide safe height of active suspension | Position Sensor + acceleration sensor (Sensing) | Signal out of range / stuck | Plausibility check using acceleration signal + Range check | Command disabled and fallback with second channel and ESC warning (communication bus off) |
| Provide safe height of active suspension | Motor (Actuation) | Provide more or less torque than desired | Compare motor torque estimation with the desired torque | Command disabled and fallback with second channel and ESC warning (communication bus off) |
| Provide safe height of active suspension | Intended function control (MCU Logic) | Freeze/logic fault | Monitor MCU Logic with External watchdog | Command disabled and fallback with second channel and ESC warning (communication bus off) |
| Provide safe height of active suspension | Communication (communication) | Freeze, data missing, delay … | Monitor Communication with E2E protocol | Command disabled and fallback with second channel and ESC warning (communication bus off) |
| Provide safe height of active suspension | Supply (Power supply) | Provide supply lower or higher than requested | Supply monitor | Supply 1 disabled and fallback with second channel and ESC warning (communication bus off) |
| Provide safe height of active suspension | Second path (channel 2) | Same faults listed for channel 1 | Monitor time of execution (not greater than EOTTI) | Command disabled and mechanical fallback (passive suspension) with vehicle dynamic managed by ESC |

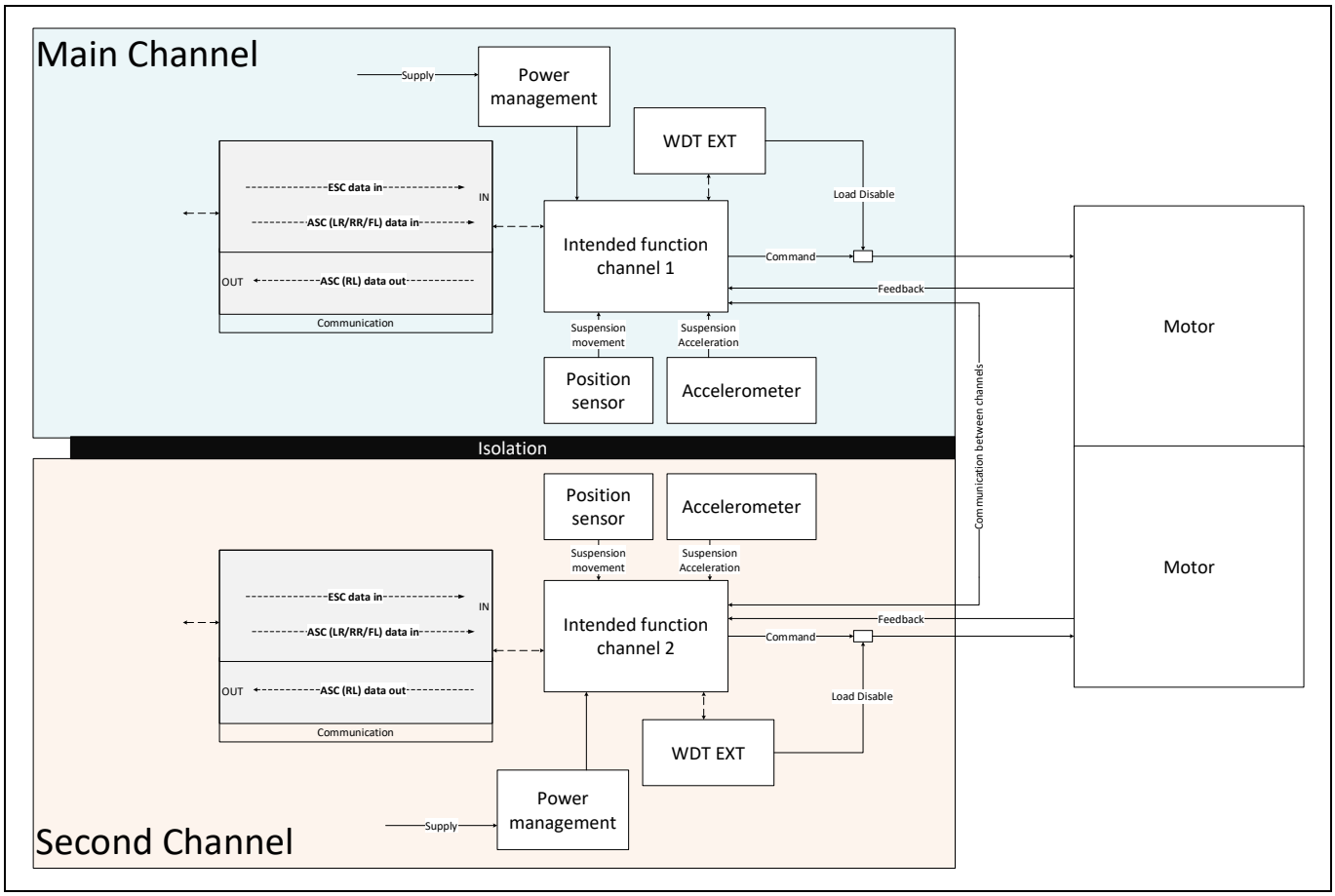



**Figure 11 Architecture layout with two independent channels (2 failsafe channels)**

## 3.3 Allocation of safety requirements

Once the system level strategy is defined, the allocation of safety requirements to hardware (HW) and software (SW) is performed as described in Table 6.

**Table 6 system monitor and allocation to HW/SW requirements**

| Element | System Monitor/Reaction | SW Requirements | HW Requirements |
|---|---|---|---|
| Position Sensor + Accelerator Sensor (Sensing) | Plausibility check with acceleration signal + Range check | MCU Logic perform plausibility check between redundant sensors and detect out of range values | Independent physical path to acquire position sensor and acceleration sensor |
| Motor (Actuation) | Compare motor torque estimation with the desired torque | MCU Logic estimates actual torque and compares with desired torque | Independence acquisition of inputs to calculate actual torque actuate + desired torque |
| Communication (communication) | Check communication periodically | MCU Logic checks E2E communication | Not applicable |

## Safety Lifecycle use case

### Product development at system level

| Element | System Monitor/Reaction | SW Requirements | HW Requirements |
|---|---|---|---|
| Intended function (MCU Logic) | External watchdog monitors the MCU Logic | MCU Logic Self-Test (LBIST) to ensure computational capabilities and Q&A with external watchdog | External watchdog to monitor independently MCU Logic (with independent clocks, supply...) |
| Supply (main path) | Supply monitor | Not applicable | HW Supply monitor (BOD, OC, OV...) |
| Second channel | Same monitor of main path | Same SW diagnostic; monitoring arbitration logic of fallback | Physically/electrically independent from main channel. No single fault shall influence both channels |

# 4 Hints for AURIX™ TC4xx family in Fail-Operational context

The concepts of redundancy and availability described in the previous section can be directly mapped to the hardware features of the AURIX™ TC4xx family, one of the most performant MCU families in Infineon portfolio. This MCU family was designed for ASIL-D systems and includes multiple safety mechanisms and fallback resources that support fail-operational architectures. The following subsections highlight the most relevant building blocks.

This subsection will be organized as follows:

- A short description of the AURIX™ TC4xx family elements
- A short description of the traditional fail-safe behavior
- A possible proposal for implementing AURIX™ TC4xx family in fail operational context

### CPU redundancy

AURIX™ TC4xx family integrates multiple TriCore™ CPUs, which are configured in lockstep mode for fault detection.

The primary fail-safe mechanism for the CPU core is the lockstep checker. In this configuration, two cores execute the same instruction stream, and their outputs are compared cycle-by-cycle. A mismatch, indicating a hardware fault, immediately triggers a CPU lockstep alarm to the Safety Management Unit (SMU). A basic fail-safe response would be for the SMU to assert an MCU reset, halting all operations to prevent the propagation of erroneous computations.

The strategy for the CPU is to isolate a core failure and redistribute the faulty core workload to other available cores, ensuring critical application functions continue to be executed. The strategy is to configure the SMU to respond to the lockstep alarm not with an MCU reset, but with a more targeted action. For any core, an interrupt can be generated to start a routine that enable a designated core to take-over. Faulty core is marked as inactive and the healthy cores takes over the critical tasks. This dynamic workload redistribution ensures that essential system functions remain available.

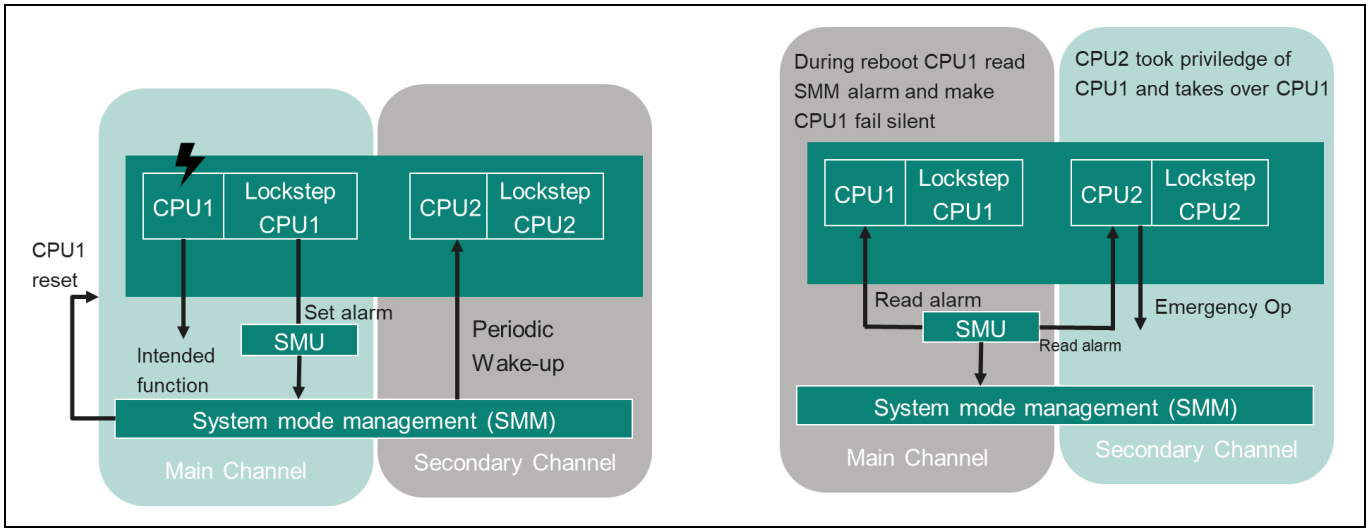



**Figure 12 Handling of CPU task handover**

### Memory protection

All the main memories are protected with Error Correction Code (ECC).

The strategy for memory is to leverage automatic hardware correction for at least single-bit error and to use software-based methods like scrubbing to manage memory degradation and uncorrectable faults, thus preserving data integrity and reducing system downtime. A built-in fail-operational feature, for example in case of single-bit errors, is the ECC. ECC logic automatically corrects the error on-the-fly without any software intervention or processing delay.

For uncorrectable errors, software can employ a redundant storage scheme, maintain a backup copy of critical data in a separate memory bank, and switch to it upon detecting a fault.

### Communication modules

The MCU offers multiple communication interfaces, such as CAN, FlexRay, Ethernet, LIN, and SPI.

A fail-safe strategy checks E2E communication to ensure the validity of the data transmitted, in case of fault the communication could be interrupted.

A fail-operational design can configure two Ethernet nodes for the same critical communication bus. If one node enters a bus-off state or suffers a hardware failure (detected via interrupt), the system software can disable the faulty node and re-route all critical messages through the second, redundant Ethernet node. This ensures no loss of critical communication with other ECUs.

### Peripheral modules

The MCU offers multiple peripheral instances of the same module.

A fail-safe reaction could lead to the peripheral module disabled.

A fault in one of the peripheral modules could be overridden by enabling another internal instance.

### Shared internal resources (power and clock)

The EVR (Embedded Voltage Regulator) includes monitoring and supervision functions to detect undervoltage or overvoltage conditions. The AURIX™ TC4xx devices also support multiple supply domains, allowing certain functions to remain powered even if one rail fails.

Monitors for critical system resources like core voltage and clock frequency are present. If a dangerous deviation is detected (e.g., overvoltage), an alarm is sent to SMU. The default fail-safe reaction is to immediately trigger the Fault Signaling Protocol (FSP) error pin, which signals an external safety device (e.g., a power supply IC) to put the entire system into a safe state (e.g., by cutting power).

SMU interrupt can trigger a reaction that places the system into a "limp-home" mode. This can be managed by a microcontroller independent in terms of supply, clock and peripheral resources.

The independent MCU can remain active even if the main TriCore™ CPUs are shut down. This independent element can perform essential functions like sending a final diagnostic message or maintaining minimal control to allow a safe stop, providing a final layer of operational capability.

# 5 Safety analysis techniques for fail-operational system

**Table 7 Safety analysis techniques for fail-operational system**

| Lifecycle phase | Analysis methods | Purpose in fail-operational context |
|---|---|---|
| Concept Phase | HARA | Safety Goals determination drives fail operational necessity |
| System Development | DFA, FTA and Sys FMEA | Assess independence and architecture robustness |
| HW Design | FMEDA, FMEA | Achieve HW target metrics and define HW fallback strategy |
| SW Design | SW FMEA, code analysis | Define SW fallback logic |
| Integration and testing | Fault injection and stress test | Demonstrate fail operational performance |
| Operation | Field return analysis | Analysis of strategy robustness |

Fail-operational system requires multiple safety analyses, which is not an exhaustive list could include:

- **HARA** to define fail operational needs
- **FTA**, **FMEDA**, **FMEA** and **DFA** to validate design robustness
- Maintaining **traceability** from safety goal to test case to **verify** all the possible failure

For SW development process, stringent analysis and tests are required. For example, a non-exhaustive list is provided as follows:

- Use **Static analysis** tools to catch runtime errors that could cause failures (buffer overflows, etc.). Ensure that the compiler and other tools are qualified or justified for use in ASIL projects
- Use **unit and integration testing** for safety SW modules. Each safety requirement can be tested via fault injection in a simulation or using hardware-in-the-loop environments. For instance, simulate a sensor failure and check the software calls the fallback routine and issues an alert within the required time
- Perform code **reviews** focusing on safety mechanism implementations to ensure they cover all cases and have no logic gaps

# 6 Related resources

- 32-bit TriCore™ AURIX™ – TC4x webpage
- AURIX™ digital documentation
- Knowledge base articles

# References

[1] ISO 26262:2018 Road vehicles- Functional safety

[2] Infineon Technologies AG: *AN1000 - FuSa in a Nutshell - release note*; Available online

## Glossary

**Table 8 Glossary**

| Definition | Description | Notes |
|---|---|---|
| AD | Automated Driving | – |
| Architectural Element | The smallest element on which the FMEDA is performed | – |
| ASC | Active Suspension Controller | – |
| ASIL | Automotive Safety Integrity Level; refer to ISO 26262-1:2018, 3.6 | – |
| BMS | Battery Management System | – |
| BOD | Brown Out Detection | – |
| CCF | Common-Cause Failure; refer to ISO 26262-1:2018, 3.18 | – |
| CPU | Central Processing Unit | – |
| DFA | Dependent Failure Analysis identifies single events that can cause multiple sub-parts to malfunction (for example, intended function and its safety mechanism) and lead to a violation of a safety requirement or safety goal. | – |
| E2E | End to End | – |
| ECC | Error Correction Code | – |
| ECU | Electronic Control Unit | – |
| EOTTI | Emergency Operation Tolerance Time Interval | – |
| ESC | Electronic Stability Control | – |
| FL | Front left Side | – |
| FMEA | Failure Mode and Effects Analysis | – |
| FMEDA | Failure Modes, Effects and Diagnostic Analysis<br>Analysis of the effect of random hardware faults on a safety requirement or safety goal, including quantitative estimation of failure rates and the probability/rate of a safety goal violation | Quantitative<br>Bottom-up<br>HW only |
| FR | Front Right Side | – |
| FSP | Fault Signal Protocol | – |
| FSR | Functional Safety Requirement | – |
| FTA | Fault Tree Analysis<br>Analysis in which a top-level failure mode is broken down to a combination of lower-level faults (root causes) using a Boolean logic approach | Qualitative (maybe quantitative)<br>Top-down<br>HW only |
| FTTI | Fault Tolerant Time Interval; for the definition refer to ISO 26262-1:2018, clause 3.61 | – |
| HARA | Hazard Analysis and Risk Assessment; Refer to ISO 26262-1:2018, 3.76 | – |

### Glossary

| Definition | Description | Notes |
|---|---|---|
| HW | Hardware | – |
| IC | Integrated Circuit | – |
| IEC | International Electrotechnical Commission | – |
| ISO | International Organization for Standardization | – |
| LBIST | Logic Built-in Self-Test | – |
| MCU | Microcontroller unit | – |
| OC | Over Current | – |
| OV | Over Voltage | – |
| Q&A | Question and Answer | – |
| RL | Rear Left Side | – |
| RR | Rear Right Side | – |
| Safety Measure | Activity or technical solution to prevent, detect, control or mitigate systematic and random failures. | – |
| SCR | Stand-by controller | – |
| SG | Safety Goal | – |
| SM | Safety Mechanism: for the definition refer to ISO 26262-1:2018, 3.142 | – |
| SMM | System Mode Management | – |
| SMU | Safety Management Unit | – |
| SS | Safe State | – |
| SW | Software | – |
| TSR | Technical Safety Requirement | – |
| WDT | Watchdog | – |

# Revision history

| Document revision | Date | Description of changes |
|---|---|---|
| V 1.0 | 2025-10-09 | Initial release |

**Trademarks**
All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2025-10-09**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2025 Infineon Technologies AG. All Rights Reserved.**

**Do you have a question about this document?**

**Email:** erratum@infineon.com

**Document reference**

**AN1106**

**Important Notice**

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability.

Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury. However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer.

Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG).

If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited. For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit https://www.infineon.com