PD ISO/PAS 19451-1:2016

BSI Standards Publication

Application of ISO 26262:2011-2012 to semiconductors

Part 1: Application of concepts

bsi.

National foreword

This Published Document is the UK implementation of ISO/PAS 19451-1:2016.

The UK participation in its preparation was entrusted to Technical Committee AUE/16, Data Communication (Road Vehicles).

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

ISBN 978 0 580 85460 6

ICS 43.040.10

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 July 2016.

Amendments issued since publication

Date	Text affected

PD ISO/PAS 19451-1:2016

PUBLICLY AVAILABLE SPECIFICATION

ISO/PAS 19451-1

First edition
2016-07-15

Application of ISO 26262:2011-2012 to semiconductors —

Part 1: Application of concepts

Application de l'ISO 26262:2011-2012 aux semi-conducteurs —

Partie 1: Application des concepts

Reference number
ISO/PAS 19451-1:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2. www.iso.org/directives

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received. www.iso.org/patents

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 22, *Road vehicles*, Subcommittee SC 32, *Electrical and electronic components and general system aspects*.

ISO/PAS 19451 consists of the following parts, under the general title *Application of ISO 26262:2011-2012 to semiconductors*:

— *Part 1: Application of concepts*

— *Part 2: Application of hardware qualification*

Introduction

This document is an informative guideline which provides users of the ISO 26262 series of standards recommendations and best practices which can be utilized when applying ISO 26262 to semiconductor components and parts. This document was created by a group of industry experts including semiconductor developers, system developers, and vehicle manufacturers in order to clarify concerns seen after the initial release of the ISO 26262 series of standards and when possible to align on common interpretations of the standard.

This document serves to augment the existing normative and informative guidance in the ISO 26262 series of standards. The approach is similar to that taken in writing ISO 26262-10:2012, Annex A, “ISO 26262 and microcontrollers,” with extension to additional types of semiconductor technologies and relevant topics.

Application of ISO 26262:2011-2012 to semiconductors —

Part 1:
Application of concepts

1 Scope

This document is applicable to developers who are evaluating the use of semiconductor components or parts in hardware components, systems, or items developed according to ISO 26262.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 26262-1, *Road vehicles — Functional safety — Part 1: Vocabulary*

ISO 26262-2:2011, *Road vehicles — Functional safety — Part 2: Management of functional safety*

ISO 26262-9:2011, *Road vehicles — Functional safety — Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 26262-1 and the following apply.

3.1
base failure rate
BFR
failure rate of a hardware element in a given application use case used as an input to functional safety analysis according to ISO 26262-5:2011, 8.4.3

3.2
guest machine
virtual instance of a *processing element* (3.7)

3.3
host machine
processing element (3.7) which implements a *hypervisor* (3.4) and one or more *guest machines* (3.2)

3.4
hypervisor
software or hardware that instantiates and manages one or more virtual design elements

Note 1 to entry: A hypervisor is sometimes referred to as a virtual machine monitor.

3.5
microkernel
μ-kernel
software which provides the minimal mechanisms needed to implement an operating system

3.6
multi-core
MC
hardware element which includes two or more hardware processing elements

3.7
processing element
PE
element providing a set of functions for data processing, normally consisting of a register set, an execution unit, and a control unit

EXAMPLE A hardware component consisting of four cores can be described as having four processing elements.

3.8
programmable logic device
PLD
device which provides user programmable logic and signal routing functions which generate application specific logic functions

3.9
virtualization
creation of a virtual (rather than physical) version of an element, including but not limited to a computer hardware platform, operating system (OS), storage device, or computer network resource

4 Symbols and abbreviated terms

ADC	Analogue to Digital Converter
ASET	Analogue Single Event Transient
BIST	Built-In Self-Test
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
DFA	Dependent Failure Analysis
DFI	Dependent Failure Initiator
DMA	Direct Memory Access
DMOS	Double Diffused Metal Oxide Semiconductor (HV MOS)
DSP	Digital Signal Processor
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
EVR	Embedded Voltage Regulator
FET	Field Effect Transistor
FFI	Freedom from Interference
FIT	Failures in Time

FMEA	Failure Modes and Effects Analysis
FPGA	Field Programmable Gate Array
FTA	Fault Tree Analysis
GPU	Graphics Processing Unit
HV	High Voltage
HW	Hardware
HS	High Side
ISA	Instruction Set Architecture
LDO	Low Drop Output Regulator
LS	Low Side
LSB	Least Significant Bit
MMU	Memory Management Unit
MPU	Memory Protection Unit
OP AMP	Operational Amplifier
OS	Operating System
OV	Over Voltage
PAL	Programmable Array Logic
PLD	Programmable Logic Device
PLL	Phase Locked Loop
RF	Radio Frequency
SEB	Single Event Burnout
SEE	Single Event Effect
SEGR	Single Event Gate Rupture
SEL	Single Event Latch-up
SET	Single Event Transient
SEU	Single Event Upset
SMPS	Switched Mode Power Supply
SoC	System on Chip
SW	Software
UV	Under Voltage
VMM	Virtual Machine Monitor

5 Analogue/mixed signal components and ISO 26262

5.1 About analogue and mixed signal components

As described in ISO 26262-10:2012, Annex A, an integrated component is structured in parts and sub-parts. If the signals that are handled in an element (component, part or sub-part) are not limited to digital states this element is seen as analogue element. This is the case for all measurement interfaces to the physical world, including sensors, actuator outputs, and power supplies.

For analogue components, all elements are analogue and no digital element is included. Mixed signal components consist of at least one analogue element and one digital element. Since analogue and digital elements require different methodologies and tooling for design, layout, verification and testing, it is recommended to clearly partition the analogue and digital blocks. The partitioning can result in a variety of configurations ranging from analogue dominated components with digital support blocks (e.g. digitally configurable voltage regulators or auto zeroing amplifiers) to microcontrollers with a few mixed signal peripherals (e.g. analogue to digital converters and phase locked loops).

A hierarchy of a typical mixed signal component including exemplary parts and sub-parts is shown in Figure 1.

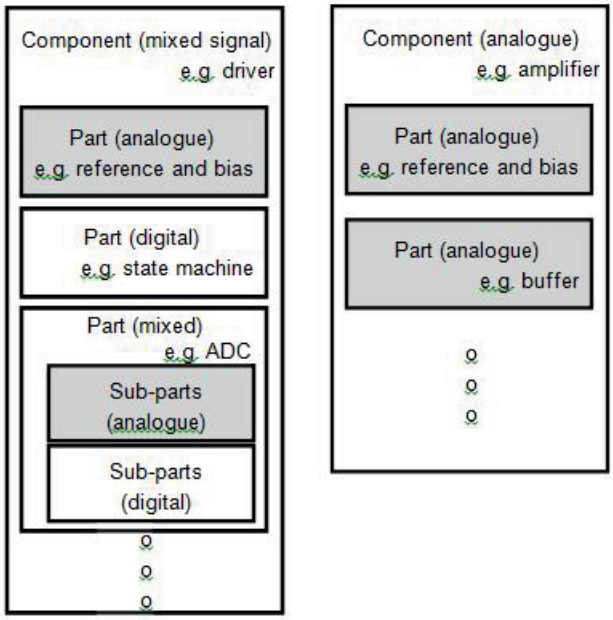

Figure 1 — Generic hierarchy of analogue and mixed signal components

It can be helpful to choose the partitioning of a mixed signal component in a way that simplifies the safety analysis. For an easy definition of fault models and failure modes, the analogue part boundaries can be defined by their function. Additionally, all elements that have freedom of interference or independence requirements (e.g. redundant paths or functions and corresponding diagnostic functions) are separated by part or sub-part boundaries. There are several additional criteria to further divide a mixed signal element (component or part) into sub elements (part or sub-part):

— Signal flow;

EXAMPLE 1 Mixed signal control loops can consist of feedback ADC, digital regulator and output driver.

— Connectivity;

EXAMPLE 2 Reference and bias circuits can serve multiple analogue blocks and oscillators can serve multiple digital or mixed signal blocks.

— Different technologies;

EXAMPLE 3 HV switch is a DMOS transistor while the gate driver can use conventional MOS devices.

NOTE One benefit for a separation of these parts is that they can have failure rates with different orders of magnitude or different fault models.

— Different supply domains;

EXAMPLE 4 Feedback DAC can be supplied with different supplies than the other mixed signal block output driver.

— Other criteria for partitioning.

EXAMPLE 5 High versus low frequency sub-parts.

The level of detail of the analysis and partitioning is determined by the relevant safety requirements, safety mechanisms and the need to show independence of safety mechanisms. A higher granularity does not necessarily result in a significant benefit for the safety analysis.

5.2 Analogue and mixed signal components and failure modes

5.2.1 About failure modes

The failure modes affecting a HW element depend on its function. The failure mode distribution depends on the HW element implementation.

NOTE The implementation includes both the actual circuit and the technology process used.

The classification of a failure mode depends on the functional and safety requirements of the system integrating the element. Based on the integration, a specific failure mode can or cannot lead to a violation of a safety requirement. Table 1 identifies possible failure modes that can be of concern for an analogue and mixed signal part or sub-part. The table can be used to extend the list of failure modes reported in ISO 26262-5:2011, Annex D.

The failure modes identified in Table 1, as well as the mentioned parts and sub-parts, are a general reference and can be adjusted on a case by case basis. The actual failure mode list used in a specific project can be adjusted (adding or removing failure modes) based on the specific implementation details or on the level of granularity deemed necessary for the analysis.

It is noted that the relevance of the failure modes, including but not limited to the ones listed in Table 1 are dependent on the context of the function to be analysed.

EXAMPLE 1 The obvious failure modes of a voltage regulator are over-voltage and under-voltage. These failure modes can be detected by an over voltage and under voltage (OV/UV) monitor as described in 5.4.2.

Besides the obvious failure modes reported in the above example, it is important to identify all relevant failure modes in order to perform a complete and thorough analysis.

EXAMPLE 2 If a voltage regulator used as a sensor supply or as an ADC reference supply, then the failure modes affecting the stability and the accuracy of the output voltage, even within the OV/UV thresholds, can be critical. Output voltage with insufficient accuracy and output voltage oscillation within the OV/UV thresholds can be mitigated by using appropriate measures. An independent ADC (internal or external) can be used to periodically measure the regulator output voltage with the required accuracy to detect those failure modes.

EXAMPLE 3 If a voltage regulator is used as a supply for a radio frequency (RF) module which has tight supply voltage ripple requirements, the prevention of fluctuation on the regulated output voltage caused by input voltage variations (i.e. the PSRR, power supply rejection ratio) is an important feature. Failure modes like output voltage oscillation within the OV/UV (i.e. ripple) limits and spikes affecting the regulated voltage can be relevant. A low pass filter as described in 5.4.8 can be used to mitigate these failures.

EXAMPLE 4 If a voltage regulator is used as an MCU core supply is sensitive to output voltage drops during start-up (power-up) due to in-rush current exceeding regulator load current and/or current limit, a too fast start-up time can be critical. A proper regulator soft-start function can be used to mitigate such failure.

If failure modes are classified as not safety related, an argument is provided in the safety analysis to support the classification.

Table 1 — Possible failure modes of analogue and mixed signal parts and sub-parts

Part/sub-part	Short description	Failure modes
Regulators and Power stages		
Voltage regulators (linear, SMPS, etc.)	HW part/sub-part that maintains the voltage of a power source within a prescribed range that can be tolerated by elements using that voltage.	Output voltage higher than a high threshold of the prescribed range (i.e. over voltage – OV) Output voltage lower than a low threshold of the prescribed range (i.e. under voltage – UV) Output voltage affected by spikes[b] Incorrect start-up time (i.e. outside the expected range) Output voltage accuracy too low, including drift[c] Output voltage oscillation[a] within the prescribed range Output voltage affected by a fast oscillation[a] outside the prescribed range but with average value within the prescribed range Quiescent current (i.e. current drawn by the regulator in order to control its internal circuitry for proper operation) exceeding the maximum value
Charge pump, regulator boost	HW part/sub-part that converts, and optionally regulates, voltages using switching technology and capacitive-energy storage elements, and maintains a constant output voltage with a varying voltage input.	Output voltage higher than a high threshold of the prescribed range (i.e. over voltage – OV) Output voltage lower than a low threshold of the prescribed range (i.e. under voltage – UV) Output voltage affected by spikes[b] Incorrect start-up time (i.e. outside the expected range) Quiescent current (i.e. current drawn by the regulator in order to control its internal circuitry for proper operation) exceeding the maximum value
High-side/Low-side (HS/LS) driver	HW part/sub-part that applies voltage to a load in a single direction: high side driver to connect the load to high rail, low side driver to connect the load to low rail.	HS/LS driver is stuck in ON or OFF state HS/LS driver is floating (i.e. open circuit, tri-stated) HS/LS driver resistance too high when turned on HS/LS driver resistance too low when turned off HS/LS driver turn-on time too fast or too slow HS/LS driver turn-off time too fast or too slow

Table 1 *(continued)*

Part/sub-part	Short description	Failure modes
Half-bridge driver or full-bridge (H-bridge) driver	HW part/sub-part that can apply voltage across a load in either direction. A half-bridge driver is built with two drivers (one HS and one LS driver). An H-bridge (or full-bridge) driver is built with four drivers (two HS and two LS drivers)	HS/LS driver is stuck in ON or OFF state HS/LS driver is floating (i.e. open circuit, tri-stated) HS/LS driver ON resistance too high when turned on HS/LS driver OFF resistance too low when turned off HS/LS driver turn-on time too fast or too slow HS/LS driver turn-off time too fast or too slow 'Dead time' is too short (i.e. when turning off high-side driver and turning on low-side driver, or when turning off low-side driver and turning on high-side driver) 'Dead time' is too long
High-side/Low-side pre-driver	HW part/sub-part driving a gate of an external FET that is used as a HS or LS driver.	HS/LS pre-driver is stuck in ON or OFF states HS/LS pre-driver output voltage/current too high or too low HS/LS pre-driver is floating (i.e. open circuit, tri-stated) HS/LS pre-driver slew rate too slow or too fast
Analogue to digital and digital to analogue converters[d]		
N bits analogue to digital converters (N-bit ADC)[d]	HW part/sub-part converting a continuous-time and continuous-amplitude analogue signal (i.e. a voltage value) to a discrete-time and discrete-amplitude digital signal coded on "N bits."	One or more outputs are stuck (i.e. high or low) One or more outputs are floating (i.e. open circuit) Accuracy error (i.e. Error exceeds the LSBs) Offset error not including stuck or floating conditions on the outputs, low resolution No monotonic conversion characteristic (i.e. given two input analogue voltage V1 > V2, the correspondent digital values are D1 < D2) Full-scale error not including stuck or floating conditions on the outputs, low resolution Linearity error with monotonic conversion curve not including stuck or floating conditions on the outputs, low resolution Incorrect settling time (i.e. outside the expected range)

Table 1 *(continued)*

Part/sub-part	Short description	Failure modes
N bits digital to analogue converters (DAC)[d]	HW part/sub-part converting digital data coded on "N bits" into an analogue signal (voltage or current).	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Offset error (not including stuck or floating conditions on the outputs, low resolution) Linearity error with monotonic conversion curve not including stuck or floating conditions on the outputs, low resolution Full-scale gain-error not including stuck or floating conditions on the outputs, low resolution No monotonic conversion curve Incorrect settling time (i.e. outside the expected range) Oscillation[a] of the output signal including drift[c]
Oscillators and clock generators		
Oscillator	HW part/sub-part generating a periodic, oscillating signal. It can be used as clock in a digital circuit.	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect output signal swing (i.e. outside the expected range) Incorrect frequency of the output signal (i.e. outside the expected range, including harmonics when applicable, for instance EMC emissions) Incorrect duty cycle of the output signal (i.e. outside the expected range) Drift[c] of the output frequency Jitter too high in the output signal
Phase locked loop (PLL)	HW part/sub-part controlling an oscillator in order to generate a square wave signal that maintains a constant phase angle (i.e. lock) on the frequency of an input, or reference signal. It can be used as clock in a digital circuit.	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect frequency of the output signal (i.e. outside the expected range, including harmonics when applicable, e.g. EMC emissions) Incorrect duty cycle of the output signal (i.e. outside the expected range) Drift[c] of the output frequency Jitter too high in the output signal Loss of lock condition (i.e. phase error, output clock not in sync with input clock not leading to incorrect frequency and incorrect duty cycle Missing pulse in the output signal Extra pulse in the output signal
Generic		

Table 1 *(continued)*

Part/sub-part	Short description	Failure modes
Voltage/Current comparator	HW part/sub-part comparing an input analogue signal with a predefined threshold (i.e. voltage or current constant value) and producing a binary signal at the output; the output depends on which is higher between the input signal and the threshold and it remains constant as the difference between them stays with the same polarity.	Voltage/Current comparator not triggering when expected Voltage/Current comparator falsely triggering Output is stuck (i.e. high or low) Output is floating (i.e. open) Oscillation[a] of the output
Operational amplifier and buffer	HW part/sub-part integrating a DC-coupled high-gain voltage amplifier with a differential input and, usually, a single-ended output.	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect gain on the output voltage (i.e. outside the expected range) Incorrect offset on the output voltage (i.e. outside the expected range) Incorrect output dynamic range (i.e. outside the expected range) Incorrect input dynamic range (i.e. outside the expected range) Output voltage accuracy too low, including drift[c] Output voltage affected by spikes[b] Output voltage oscillation[a] Settling time of the output voltage too low
Sample and hold	HW part/sub-part sampling the voltage of a continuously varying analogue input signal and holding its value at a constant level for a specified minimum period of time.	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect sampling leading to gain/offset error on output voltage dependent on input signal Incorrect gain on the output voltage (i.e. outside the expected range) Incorrect offset on the output voltage (i.e. outside the expected range) Incorrect output dynamic range (i.e. outside the expected range) Incorrect input dynamic range (i.e. outside the expected range) Output voltage accuracy too low during hold phase, including drift[c] Output voltage during hold phase affected by spikes[b] Output voltage oscillation[a] during hold phase Output does not settle sufficiently accurate during hold time

Table 1 *(continued)*

Part/sub-part	Short description	Failure modes
Analogue multiplexer	HW part/sub-part consisting of multiple analogue input signals, multiple control inputs and one output signal.	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect channel selection Offset affecting the output signal too high Resistive or capacitive coupling among input channels and output signal including crosstalk Resistive or capacitive coupling among selectors and output signal including crosstalk Incorrect output dynamic range (i.e. outside the expected range) Attenuation of the output signal Drift[c] affecting the output signal Spikes[b] affecting the output signal (i.e. during switching)
Analogue switch	HW part/sub-part capable of switching or routing analogue signals based on the level of a digital control signal. Commonly implemented using a "transmission gate".	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit or tri-stated) Offset too high affecting the output signal Resistive or capacitive coupling between control signal and output signal including crosstalk Attenuation of the output signal Drift[c] affecting the output signal Spikes[b] affecting the output signal, e.g. during switching
Passive network	HW part/sub-part consisting of a network of passive devices (resistor and capacitor) providing a specific low pass transfer function	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect output dynamic range (i.e. outside the expected range) Incorrect attenuation of the output signal (i.e. outside the expected range) Incorrect settling time (i.e. outside the expected range) Drift[c] affecting the output signal Oscillation[a] affecting the output signal (i.e. due to crosstalk, coupling or parasitic effects) Spikes[b] affecting the output (i.e. due to crosstalk, coupling or parasitic effects)

Table 1 *(continued)*

Part/sub-part	Short description	Failure modes
Voltage references	HW part/sub-part producing a constant DC (direct-current) output voltage regardless of variations in external conditions such as temperature, barometric pressure, humidity, current demand, or the passage of time.	Output is stuck (i.e. high or low) Output is floating (i.e. open circuit) Incorrect output voltage value (i.e. outside the expected range) Output voltage accuracy too low, including drift[c] Output voltage affected by spikes[b] Output voltage oscillation[a] within the expected range Incorrect start-up time (i.e. outside the expected range)
Current source (including bias current generator)	HW part/sub-part delivering or absorbing a current (i.e. reference current) which is independent of the voltage across it. It typically includes multiple branches which are routed to other circuits requiring a reference or bias current.	One or more outputs are stuck (i.e. high or low) One or more outputs are floating (i.e. open circuit) Incorrect reference current (i.e. outside the expected range) Reference current accuracy too low, including drift[c] Reference current affected by spikes[b] Reference current oscillation[a] within the expected range One or more branch currents outside the expected range while reference current is correct One or more branch currents accuracy too low, including drift[c] One or more branch currents affected by spikes[b] One or more branch currents oscillation[a] within the expected range

Table 1 *(continued)*

Part/sub-part	Short description	Failure modes
a An oscillation is an instability of the part/sub-part caused by internal failure, e.g. regulation loop failures, lower or negative hysteresis for a comparator, etc.. Oscillation includes any repetitive voltage and current variation (i.e. periodic pulse) b A spike is a non-repetitive variation on the output voltage or current, i.e. pulse due to load jumps, etc. c Drift is a slow and continuous variation of a parameter (i.e. current, voltage, threshold, etc.) outside the expected range reported into the circuit specification. Slow variation means slower as compared to the fault tolerant time interval (FTTI). For example drift covers floating or stuck at open failure modes. d Several of the failure modes reported for the ADC or DAC can be grouped into two main sets: static error and absolute accuracy (total) error Static errors are errors that affect the accuracy of a converter when it is converting static (DC) signals and can be completely described by four terms: — offset error; — gain error; — integral non-linearity; — differential non-linearity. Each term can be expressed in LSB units or sometimes as a percentage of the full scale range (FSR). For example, an error of 1/2 LSB for an 8-bit converter corresponds to 0,2 % FSR. The absolute accuracy (total) error is the maximum value of the difference between an analogue value and the ideal mid-step value. It includes offset, gain, and integral linearity errors, and also the quantization error in the case of an ADC.		

Given the variety of implementations and the lack of data available from the field and from theory, Table 1 does not give any indication about the quantitative impact of the listed failure modes, i.e. the failure mode distribution. It is the responsibility of the safety analyst to identify such quantitative data. An example is given in 5.3.3.

NOTE 1 Even though it is known that a single physical root cause can lead to more than one failure mode, it is a common simplification that the sum of the distribution of all failure modes is 100 %.

NOTE 2 Transient failure modes are considered if they are relevant, for example if for the technology in use the risk of single-event effects (SEE) is not negligible, see 5.2.3.

5.2.2 About safe faults

ISO 26262-10:2012, 8.1.7 states that safe faults can be faults of one of two categories:

— all n point faults with $n > 2$, unless the safety concept shows them to be a relevant contributor to a safety requirement, or

— faults that will not contribute to the violation of a safety requirement.

Analogue components are characterized by continuous (output) signal (function) regions and as such, tolerances shall be taken into consideration when used in systems. The tolerances on analogue functions as specified as part of the safety requirements allocated to that analogue component can be less constrained than the actual tolerance of the analogue component itself. For this reason, the fraction of the failure mode that leads to parametric failure or drift, but which remains within these tolerance ranges is safe. An analogue component has therefore an inherent capability to tolerate a fault. These faults are safe faults.

EXAMPLE 1 A resistor is used to limit the current flowing through a specific branch. A failure in the accuracy of the resistor increasing its value (e.g. of 50 %) but not preventing the current limiting function would be a safe fault.

A specific fault in an element can have a different classification depending on the specific safety requirement considered. For more details see ISO 26262-5.

Depending on the system configuration and the system safety requirements some failure modes are not relevant, i.e. they cannot violate the requirements. In this case, these failure modes can be classified as safe: They contribute to the safety metrics increasing the failure rate of safe faults

EXAMPLE 2 An output driver can have an output slope control to limit the rise and fall times of the output value for EMI purposes. If the slew rate is irrelevant for the violation of the safety goal, failures in this slope control would be safe faults.

EXAMPLE 3 If a voltage regulator is used to supply digital circuits only, failure modes affecting the stability and the accuracy of the output voltage within the OV/UV thresholds can be classified as safe.

5.2.3 About transient faults

As defined in ISO 26262-1:2011, definition 1.135, a transient fault is a fault that occurs once and subsequently disappears. Soft errors such as Single Event Upset (SEU) and Single Event Transient (SET) are defined as transient faults. ISO 26262-5:2011, 8.4.7 states that transient faults are considered when shown to be relevant due, for instance, to the technology used and can be addressed either by a quantitative approach, specifying and verifying a dedicated target “single-point fault metric” value to them or by a qualitative rationale based on the verification of the effectiveness of the internal safety mechanisms implemented to cover these transient faults.

In terrestrial analogue circuits, transient faults are caused by alpha-particle or neutron hits or by electromagnetic interference such as power transients and crosstalk. They can cause SEU or even SET also called Analogue Single Event Transients (ASETs), such as transient pulses in operational amplifiers, comparators or reference voltage circuits.

Due to the intrinsic nature of analogue technology (in which transient or noise effects are considered by design), the susceptibility to transient faults is lower than in digital circuits by orders of magnitude. Therefore, the analysis of those effects can be limited in a first approximation to their digital part (e.g. the digital decimation filter of a sigma-delta ADC).

However in some cases, like in the early part of the conversion cycle of an ADC (see reference[37]) or in PLL (see reference[28]) or differential switched-capacitor circuits (see reference[17]), the vulnerability to soft-error can be high. In those cases, more detailed analyses are done and appropriate countermeasures are identified (see reference[8]).

For mixed signal components, the impact of soft errors in the digital part is considered as described in ISO 26262-5. ISO 26262-10 can also be referred as a guideline.

NOTE If more detailed analyses are needed in the analogue part, since SER evaluation by irradiation tests in analogue circuits is not a simple task, in those cases measurement is done mainly by analytical.

5.3 Notes about safety analysis

5.3.1 General

The examples and guidelines given in ISO 26262-10:2012, Annex A are also valid for an analogue or mixed signal component. The following paragraphs describe some of the topics that can require additional clarification for an analogue or mixed signal component.

5.3.2 Level of granularity of analysis

One of the key aspects for the safety analysis of analogue elements is the proper identification of the level of hierarchy on which to base the analysis. On one hand, a lower level of granularity is beneficial as it allows for a better understanding of the failure modes and failure mode distributions. On the other, a higher level of granularity allows for a clear allocation of safety mechanisms. Analogue elements are

often used to interface with physical objects making it useful to also consider mechanical characteristics and differentiate the failure modes accordingly.

As seen in ISO 26262-9:2011, 8.2, qualitative and quantitative safety analyses are performed at the appropriate level of abstraction during the concept and product development phases. The level of abstraction can be consequently adjusted depending on the target of the analysis. Qualitative analysis is more suited to identify failure modes while quantitative analysis to quantify their failure rate and distribution.

Consider an example in which, a linear voltage regulator is monitored using a windowed voltage monitor. The voltage monitor is at the output of the regulator and is able to detect over-voltage conditions. If the output value, allowed in the working condition to fluctuate in a range around a nominal value, e.g. 1,2 V ± 0,12 V, moves outside that range it is to be considered faulty. If the analysis focuses on the output of the regulator it can be relatively easy to discriminate between types of failures (e.g. safe because within allowed range, over or under voltage) and quantify the protection offered by the voltage monitor. However it is difficult to quantify the likelihood of each type of failure as required for metric computation. If the analysis goes inside the regulator and focuses, for instance, on faults of the bandgap it is easier to analyse propagation and likelihood of each failure of the regulator but not simple to quantify the protection that the external voltage monitor offers on the bandgap itself.

For the safety analysis, the type of safety mechanisms can drive the selection of the level of abstraction. If the safety mechanisms addressing analogue features are located at system level, going down in the block structure can lead to an overly complex analysis. The quantification of the failure mode distribution can require an investigation on lower levels of abstraction. For instance, applying an equal distribution to the failure modes of the linear voltage regulator can give less accurate results than applying an equal distribution to the blocks composing the linear voltage regulator as, for instance, the bandgap, the buffer, the driver, etc. With respect to terminology, in line with the classification done for microcontroller in ISO 26262-10:2012, Annex A and according to ISO 26262-10:2012, Table D.1, the linear voltage regulator is to be considered a part and the bandgap, the buffer, the driver, etc. sub-parts.

5.3.3 Examples of usage of failure mode distributions

The failure distribution model is dependent on the circuit implementation and targeted process. Each supplier provides details on the failure mode distribution model used in the analysis.

EXAMPLE 1 A simple and pessimistic model can be used for the initial analysis, like considering only failure modes capable of violating a safety requirement (i.e. not a safe failure mode) and using a linear distribution for the defined failure modes; for instance if five failure modes are defined, each failure mode is allocated 20 % distribution.

NOTE 1 In the EXAMPLE 1 above, this analysis considers all the applicable failure modes except those not capable of violating the safety requirement. Safe failure modes are not included in the computation.

If the analysis using such failure mode distribution model does not fulfil the required Single Point Fault and/or Latent Fault metrics for the targeted ASIL level, the definition of the failure modes and related distribution is further refined.

EXAMPLE 2 Failure modes not capable of violating the safety requirement, i.e. safe failure modes, that are applicable to the circuit under analysis, are added in the computation with F_{safe} = 100 %

NOTE 2 The uniform failure mode distribution and the list of safe and not safe failure modes are considered in the FMEA example in 5.3.5.

EXAMPLE 3 A more detailed distribution for all failure modes can be considered based on area; if the area of the circuit or circuits identified as the root cause for the defined failure mode is 5 %, then the allocated failure mode distribution is 5 %.

Applicable failure modes and detailed failure mode distributions are justified according to the circuit implementation and its physical area and documented in the product safety case.

5.3.4 Example of failure rates estimation for an analogue part

Calculation methods to derive the base failure rate for analogue and mixed signal components are described in Clause 9.

The base failure rate is allocated to the different elements composing the hardware component. Different allocation methods can be applied depending on the type of elements considered.

The base failure rate can be considered proportional to the area of the circuit.

EXAMPLE 1 The base failure rate divided by the overall area of the component in order to obtain FIT/mm^2 for each relevant fault model as reported in Table 2.

Table 2 — Base failure rate allocation based on area

Fault model	Failure rate value	Unit
Permanent faults	2,00E-02	FIT/mm^2
Transient faults	2,00E-05	FIT/mm^2

The failure rate of each sub-part of the analogue and mixed signal component shown in Figure 2 is computed by using the FIT/mm^2 reported in Table 2.

The results of the computation, considering the block diagrams of Figure 2, Figure 3, and Figure 4 are reported in Table 3.

Table 3 — Failure rate for each part/sub-part

Part	Sub-part	Block Area mm^2	Failure rate Permanent faults FIT	Failure rate Transient faults FIT
Low Drop Regulator	Regulator Core	0,52	0,0104	0,0000104
	BANDGAP 1	0,15	0,0030	0,0000 030
	Bias Current Generator	0,01	0,0002	0,0000002
	Current Limiter	0,075	0,0015	0,0000015
	TOTAL	**0,755**	**0,0151**	**0,000 0151**
Voltage Monitor	CMP1	0,03	0,0006	0,0000006
	CMP2	0,03	0,0006	0,0000006
	Passive Network	0,08	0,0016	0,0000016
	BANDGAP 2	0,15	0,0030	0,0000030
	TOTAL	**0,29**	**0,0058**	**0,0000058**
ADC	ADC	0,85	0,0170	0,0000170
Analogue BIST	Analogue BIST	0,35	0,0070	0,0000070
	TOTAL	**2,535**	**0,0507**	**0,0000507**

NOTE 1 The numbers reported in Table 2 and Table 3 are only examples.

NOTE 2 Block area reported in Figure 3 includes internal routing. Routing at top level, if relevant, is included in a separate block.

In alternative to the area-based approach, as seen in ISO 26262-10:2012, A.3.3.1, the failure rate and failure mode distribution can be estimated based on the number of equivalent transistors for each sub-part or elementary part. In the case of mixed signal or analogue components, distinction between active devices, passive devices and routing can be taken into account in the estimation of the number of equivalent transistors. The selection of the method used can be based on the layout (or planned

layout) of the circuit under analysis or on the analysis of how failure modes are shared between the HW elements.

NOTE 3 For a transient fault model, the base failure rate proportional to area is a simplified example because, in reality, not all the elements in a mixed signal circuit have the probability of failure.

EXAMPLE 2 In switched-capacitor architectures, the capacitors holding the signal are more sensitive with respect to transient faults than other portions of the circuit because they are used as memory elements.

5.3.5 Example of safety metrics computation

The following is an example of a quantitative analysis using the method described in ISO 26262-10:2012, A.3.3 in order to calculate the single-point fault metric and the latent-fault metric for a given safety requirement allocated to the mixed signal HW element depicted in Figure 2.

The example consists of a mixed signal HW element composed of:

— a low drop voltage regulator (low drop voltage regulator in Figure 3) providing an output voltage within a prescribed range;

— a voltage monitor (voltage monitor in Figure 4) capable of detecting overvoltage (VA > OV_th) and under-voltage (VA < UV_th) on the LDO output by monitoring the regulated voltage VA and comparing it with two predefined thresholds; the predefined thresholds are generated from a reference voltage provided by an independent bandgap (bandgap2 in Figure 4) in order to ensure independence with respect to the voltage regulator;

— an analogue BIST controlled through the digital system (the digital controller is not depicted in the block diagram in Figure 2);

— an ADC channel.

The ASIL B safety requirement is: “The regulated voltage output does not go out of regulation, i.e. the regulated voltage VA is not outside the VA_UV-VA_OV range for more than 1ms.”

The component can be considered in a safe state when an out of regulation condition is detected and signalled to an external element of the system/item. The external system is responsible for fault reaction including transitioning the system to a safe state.

As shown in Figure 4, the voltage monitor is composed of two voltage comparators, a passive network and a bandgap; the low drop regulator includes a bandgap, a current limiter, the bias generator and the regulator core as shown in Figure 3.

The ADC is included in the mixed signal HW element but it is not used for any function related to the safety requirement and so its potential failure cannot contribute to the violation of such requirement; therefore the ADC is assumed not safety related.

NOTE 1 The example shows that parts which could be easily isolated and disabled in a way that they can be considered not safety-related without risk, can coexist with parts that are safety related.

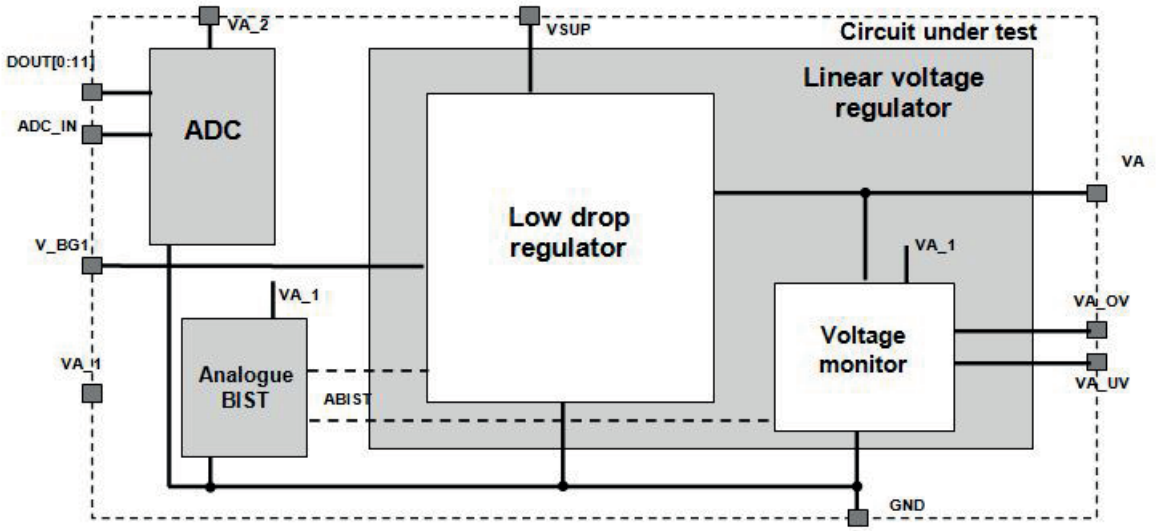

Figure 2 — Example of analogue and mixed signal HW element (circuit under analysis)

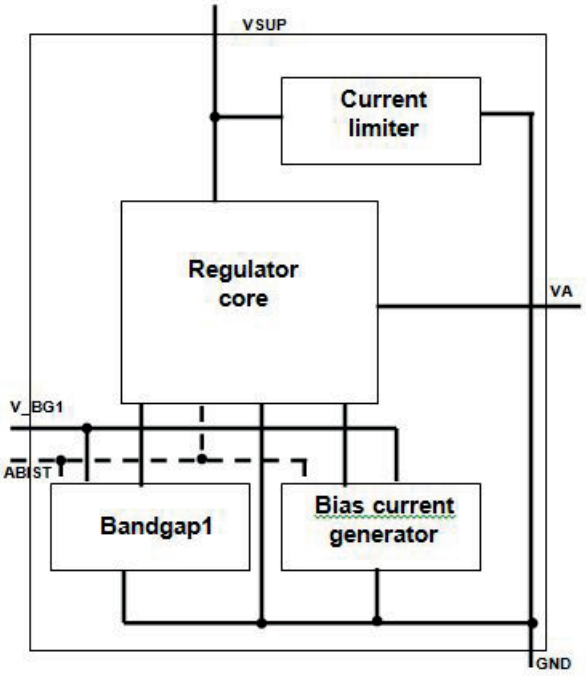

Figure 3 — Detailed block diagram of the low drop regulator part

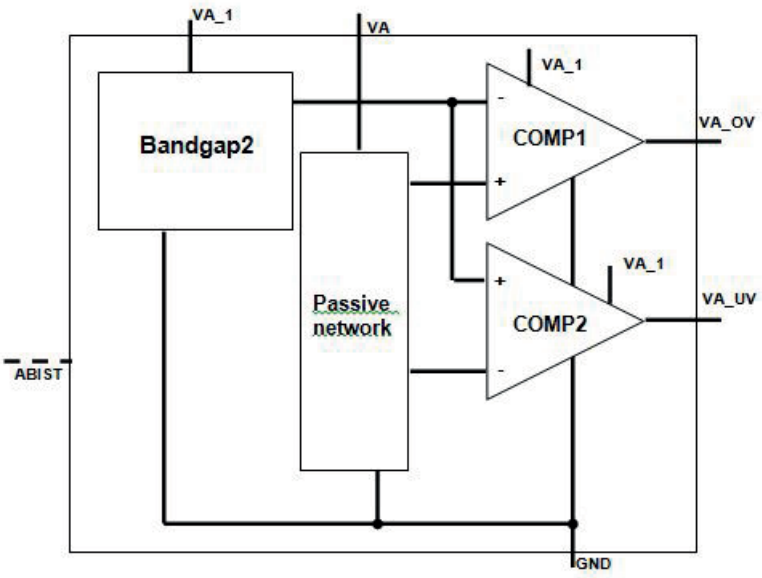

Figure 4 — Detailed block diagram of the voltage monitor part

The following safety mechanisms are considered:

— The voltage monitor detecting overvoltage (safety mechanism SM2) and under-voltage (safety mechanism SM1) failures with a diagnostic coverage of 99,9 %. The safety mechanism is described in 5.4.2.

— the analogue BIST detecting failures affecting the voltage monitor with a diagnostic coverage of 60 % (safety mechanism SM6). The safety mechanism is described in 5.4.10.

The coverage levels claimed by the safety mechanisms are reported in the following Table 4. They are assumed to be proven with simulations, testing to characterize and confirm the behaviour of the silicon and the related evidences are documented in the product safety case. It is out of the scope of this example to provide those evidences.

Each safety mechanism signals the detection of a fault to an element of the system/item which is then responsible to transition the system to a safe state.

Under this assumption, the failure mode coverage with respect to latent failures related to the low drop regulator is claimed to be 100 % based on the example in ISO 26262-5:2011, Annex E.

Table 4 — Safety mechanisms considered in the example and related coverage for HW element

ID	Safety mechanism	Claimed failure mode coverage
SM1	Under-voltage (UV) Monitor	99,9 %
SM2	Over-voltage (OV) Monitor	99,9 %
SM6	Analogue BIST diagnostics	60 %

NOTE 2 The effectiveness of safety mechanisms could be affected by dependent failures. Adequate measures are considered as described in 5.3.6.

Based on the guidelines provided in ISO 26262-10:2012, A.3.3.1, the failure rates and the metrics can be computed in the following way for analogue and mixed signal HW elements:

— First, the HW element is divided into parts or sub-parts;

NOTE 3 The validity of assumptions on the independence of identified parts is established during the dependent failure analysis.

NOTE 4 The necessary level of detail (e.g. if analysis at part level or sub-part level) can depend on the stage of the analysis and on the safety mechanisms.

— Second, the failure rates of each part or sub-part can be computed using one of the methods described in 5.3.4;

NOTE 5 In this example the failure rate distribution is assumed to be proportional to the area both for permanent and transient faults using the values reported in Table 2.

— For each part/sub-part the relevant failure modes are listed and a failure mode distribution is assigned to each of them;

The failure mode distribution in the examples of Table 5 and Table 6 is considered equally distributed over the failure modes belonging to each part/sub-part. This assumption is to be understood as reference only, valid for the specific examples.

— The evaluation is completed by classifying the faults into safe faults, residual faults, detected dual-point faults and latent dual-point faults;

— Finally, the failure mode coverage with respect to residual and latent faults of that part or sub-part is determined.

NOTE 6 Numbers used in this example (e.g. failure rates, amount of safe faults and failure mode coverage) can vary from architecture to architecture.

The example of quantitative analysis, limited to permanent faults, is reported in Table 5 and Table 6 using the same format of ISO 26262-10:2012, Table A.5. The quantitative analysis gives the view of failure modes at sub-part level.

NOTE 7 In this example a separate analysis with respect to transient faults is not reported but it can be added when relevant.

Depending on the system functions and safety requirements, different operating phases can be relevant and so additional failure modes can be considered.

EXAMPLE For systems that need to fulfil start-stop requirements, the regulator start phase can be safety relevant and the failure mode “Incorrect start-up time (i.e. outside the expected range) - Voltage ramp too fast” can be added.

Table 5 — Example of quantitative analysis – mission parts

Part	Sub-part	Safety Related Component or No Safety Related Component	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/ FIT
Low Drop Regulator	Low Drop Regulator	SR	Regulated voltage higher than VA_OV	P	14 %	2,16E-03	0 %	SM2	99,9 %	2,16E-06	SM2	100 %	0,0E+00
		SR	Regulated voltage lower than VA_UV	P	14 %	2,16E-03	0 %	SM1	99,9 %	2,16E-06	SM1	100 %	0,0E+00
		SR	Regulated voltage out of the expected range (VA_ UV-VA_OV)	P	14 %	2,16E-03	0 %	SM1 SM2	99,9 %	2,16E-06	SM1 SM2	100 %	0,0E+00
		SR	No effect-Regulated voltage within the expected range but with low accuracy	P	14 %	2,16E-03	100 %			0,0E+00			0,0E+00
		SR	Regulated voltage out of the expected range (VA_ UV-VA_OV)	P	14 %	2,16E-03	0 %	SM1 SM2	99,9 %	2,16E-06	SM1 SM2	100 %	0,0E+00
		SR	No effect-Regulated voltage within the expected range but with low accuracy	P	14 %	2,16E-03	100 %			0,0E+00			0,0E+00
		SR	no effect - assuming during the voltage regulator start up the item is in safe state	P	0 %	0,00E+00	100 %			0,00E+00			0,0E+00

Table 5 *(continued)*

Part	Sub-part	Safety Related Component or No Safety Related Component	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/ FIT
		SR	no effect - assuming during the voltage regulator start up the item is in safe state	P	0 %	0,00E+00	100 %			0,00E+00			0,0E+00
		SR	Regulated voltage potentially with low accuracy or out of regulation depending on the actual quiescent current	P	14 %	2,16E-03	50 %			1,08E-03	SM1 SM2	100 %	0,0E+00
ADC	ADC	NSR		P	100 %	7,00E-03				0,0E+00			0,0E+00
						Σ					0,00109		
						Total failure rate	0,0221						
						Total Safety Related	0,0151						
						Total Not Safety Related	0,0070						
										Single Point Faults Metric	92,8 %		
											Latent Faults Metric		100 %

[a] Depending on complexity it can be beneficial to have a dedicated entry in the FMEA giving more details about the potential root causes and the end effect of each failure mode.

[b] Fault model can be permanent fault (P) or transient fault (T); the example is limited to permanent faults.

Table 6 — Example of quantitative analysis – safety mechanisms

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
Voltage Monitor (SM1, SM2)	Voltage Monitor (SM1, SM2)	SR	UV Monitor (SM1) falsely triggering UV event	Nuisance shutdown at nominal regulator loads	P	25 %	1,45E-03	100 %						0,0E+00
		SR	UV Monitor (SM1) not triggering valid UV event	Regulated voltage lower than VA_UV	P	25 %	1,45E-03	0 %				SM6	60 %	5,80E-04
		SR	OV Monitor (SM2) falsely triggering OV event	Nuisance shutdown at nominal regulator loads.	P	25%	1,45E-03	100%						0,0E+00

Table 6 *(continued)*

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
		SR	OV Monitor (SM2) not triggering valid OV event	Regulated voltage higher than VA_OV	P	25%	1,45E-03	0%				SM6	60%	5,80E-04
Analog BIST	Analog BIST (SM6)	SR	Analogue BIST (SM6) falsely detects misbehaviour of the linear voltage regulator	No effect[c]	P	50%	3,50E-03	100%						0,0E+00
		SR	Analogue BIST (SM6) does not detect misbehaviour of the linear voltage regulator	No effect[c]	P	50%	3,50E-03	100%						0,0E+00
					Σ						0,00109			1.16E-03
					Total failure rate 0,01280									
					Total Safety Related 0,01280									
					Total Not Safety Related 0,00000									
											Single Point Faults Metric 100%			
													Latent Faults Metric 90.0%	

[a] Depending on complexity it can be beneficial to have a dedicated entry in the FMEA giving more details about the potential root causes and the end effect of each failure mode.

[b] Fault model can be permanent fault (P) or transient fault (T); the example is limited to permanent faults.

[c] It requires more than two faults before it becomes safety relevant: #1 fault: Failure on main safety mechanism (SM1, SM2 or SM3), #2 fault: LDO out of regulation, #3 fault: BIST Diagnostic failure.

Combining together the results of Table 5 and Table 6, the overall values are:

— Single Point Faults Metric = 96,1 %;

— Latent Faults Metric = 95,7 %.

The following example considers the same HW element reported in Figure 2 but a different safety requirement: "The accuracy and the stability of the regulated voltage is such that VA < VA0 + Δ and VA > VA0-Δ where VA0 is within V_{min} – V_{max} and Δ = 5mV."

The component can be considered in a safe state when the low accuracy/stability condition is detected and signalled to an external element of the system/item. The external system is responsible for fault reaction including transitioning the system to a safe state.

The example of quantitative analysis limited to permanent faults is reported in Table 8 using the same format of ISO 26262-10:2012, Table A.5. The safety mechanisms considered in the analysis are:

— The voltage monitor (voltage monitor in Figure 4) detecting overvoltage (safety mechanism SM2) and under-voltage (safety mechanism SM1) failures.

— The independent ADC channel detecting variation of the regulated voltage higher than Δ = 5mV (safety mechanism SM3). The safety mechanism is described in 5.4.11.

— A current limiter detecting failures affecting circuits supplied by the low drop voltage (safety mechanism SM5). The safety mechanism is described in 5.4.5.

— An analogue BIST detecting failures affecting the voltage monitor.

NOTE 8 The independence of the current limiter with respect to the regulator core will be evaluated in the safety analysis.

NOTE 9 The ADC used as safety mechanism SM3 is assumed to be external to the HW element under analysis and so it is not considered in the FMEA. There is an ADC included in the HW element which is not SM3: It is therefore reported in the FMEA as not safety related.

The coverage levels claimed by the safety mechanisms are reported in the following Table 7.

Table 7 — Safety mechanisms considered in the example with the new safety requirement

ID	Safety mechanism	Claimed failure mode coverage
SM1	Under-voltage (UV) Monitor	99,9 %
SM2	Over-voltage (OV) Monitor	99,9 %
SM3	Independent ADC monitoring	97 %
SM5	Current limiter	98 %
SM6	Analogue BIST diagnostics	90 %

NOTE 10 The effectiveness of safety mechanisms could be affected by dependent failures. Adequate measures are considered as described in 5.3.6.

Moreover, each safety mechanism signals the detection of a fault to an external element of the system/item which is then responsible to transition the system to a safe state.

Under this assumption, the failure mode coverage with respect to latent failures related to the mission circuit is claimed to be 100 % according to ISO 26262-5:2011, Annex E.

Table 8 shows the quantitative analysis for the mission part conducted at a finer level of granularity than the one in Table 5 and Table 6. The examples show that a different safety requirement impacts the level of partitioning and the diagnostic coverage requirement for one or more safety mechanisms.

NOTE 11 In this example the analysis with respect to transient faults is not reported but it can be added when relevant.

Table 8 — Example of quantitative analysis in case of fine granularity – mission parts

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/ FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
Linear voltage regulator	Regulator core	SR	Output voltage higher than a predefined high threshold of the prescribed range (i.e. Over voltage – OV)	Regulated voltage higher than VA_OV	P	14 %	1,49E-03	0 %	SM2	99,9 %	1,49E-06	SM2	100 %	0,00E+00
		SR	Output voltage lower than a predefined low threshold of the prescribed range (i.e. Under voltage – UV)	Regulated voltage lower than VA_UV	P	14%	1,49E-03	0%	SM1	99,9%	1,49E-06	SM1	100%	0,00E+00
		SR	Output voltage affected by spikes	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	14%	1,49E-03	0%	SM1 SM2	99,9%	1,49E-06	SM1 SM2	100%	0,00E+00
		SR	Output voltage oscillation within the prescribed range	Regulated voltage within the expected range but with low accuracy	P	14%	1,49E-03	0%	SM3	97,0%	4,46E-05	SM3	100%	0,00E+00
		SR	Output voltage fast oscillation outside the prescribed range but with average value within the prescribed range	Regulated voltage within the expected range but with low accuracy	P	14%	1,49E-03	0%	SM1 SM2	99,9%	1,49E-06	SM1 SM2	100%	0,00E+00
		SR	Output voltage drift within the prescribed range	Regulated voltage within the expected range but with low accuracy	P	14%	1,49E-03	0%	SM3	97,0%	4,46E-05	SM3	100%	0,00E+00

Table 8 *(continued)*

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/ FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
		SR	Quiescent current (i.e. current drawn by the regulator in order to control its internal circuitry for proper operation) exceeding the maximum value	Regulated voltage potentially with low accuracy depending on the actual quiescent current	P	14%	1,49E-03	50%	SM3	97,0%	2,23E-05	SM3	100%	0,00E+00
	Bandgap 1	SR	Output is stuck (high or low)	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	20%	6,00E-04	0%	SM1 SM2	99,9%	6,00E-07	SM1 SM2	100%	0,00E+00
		SR	Output is floating (e.g. open circuit)	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	20%	6,00E-04	0%	SM1 SM2	99,9%	6,00E-07	SM1 SM2	100%	0,00E+00
		SR	Output voltage oscillation within the expected range	Regulated voltage within the expected range but with low accuracy	P	20%	6,00E-04	0%	SM3	97,0%	1,80E-05	SM3	100%	0,00E+00

Table 8 *(continued)*

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/ FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
		SR	Incorrect output voltage value (i.e. outside the expected range)	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	20%	6,00E-04	0%	SM1 SM2	99,9%	6,00E-07	SM1 SM2	100%	0,00E+00
		SR	Output voltage accuracy too low, including drift	Regulated voltage within the expected range but with low accuracy	P	20%	6,00E-04	50%	SM3	97,0%	9,00E-06	SM3	100%	0,00E+00
		SR	Output voltage affected by spikes	Not applicable due to circuit implementation	P	0%	0,00E+00	0%			0,00E+00	SM1 SM2	100%	0,00E+00
	Bias current generator	SR	One or more outputs are stuck (high or low)	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	10%	2,00E-05	0%	SM1 SM2	99,9%	2,00E-08	SM1 SM2	100%	0,00E+00
		SR	One or more outputs are floating (e.g. open circuit)	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	10%	2,00E-05	0%	SM1 SM2	99,9%	2,00E-08	SM1 SM2	100%	0,00E+00
		SR	Incorrect reference current (i.e. outside the expected range)	Regulated voltage out of the expected range (VA_UV-VA_OV)	P	10%	2,00E-05	0%	SM1 SM2	99,9%	2,00E-08	SM1 SM2	100%	0,00E+00
		SR	Reference current accuracy too low, including drift	Regulated voltage within the expected range but with low accuracy	P	10%	2,00E-05	0%	SM3	97,0%	6,00E-07	SM3	100%	0,00E+00

Table 8 *(continued)*

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/ FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
		SR	Reference current affected by spikes	Regulated voltage with low accuracy for a limited time period if the spike is not filtered out; No effect otherwise	P	10%	2,00E-05	50%			1,00E-05	SM1 SM2	100%	0,00E+00
		SR	Reference current oscillation within the expected range	Regulated voltage within the expected range but with low accuracy	P	10%	2,00E-05	0%	SM3	97,0%	6,00E-07	SM3	100%	0,00E+00
		SR	One or more bias currents outside the expected range while reference current is correct	Regulated voltage with low accuracy or out of regulation	P	10%	2,00E-05	0%	SM3	97,0%	6,00E-07	SM3	100%	0,00E+00
		SR	One or more bias currents accuracy too low , including drift	Regulated voltage within the expected range but with low accuracy	P	10%	2,00E-05	0%	SM3	97,0%	6,00E-07	SM3	100%	0,00E+00
		SR	One or more bias currents affected by spikes	Regulated voltage with low accuracy for a limited time period if the spike is not filtered out; No effect otherwise	P	10%	2,00E-05	50%			1,00E-05	SM1 SM2	100%	0,00E+00

Table 8 *(continued)*

Part	Sub-part	Safety Related Component or No Safety Related Component	Failure Mode	Potential Effect of Failure Mode in Absence of Safety Mechanism (SM) on IC level[a]	Fault Model[b]	Failure distribution	Failure rate (FIT)	Amount of Safe Faults	Safety mechanism(s) preventing the violation of the safety requirement	Failure mode coverage with respect to violation of safety requirement	Residual or Single Point Fault failure rate/ FIT	Safety mechanism(s) to prevent latent faults	Failure mode coverage with respect to Latent failures	Latent Multiple Point Fault failure rate/FIT
		SR	One or more bias currents oscillation within the expected range	Regulated voltage within the expected range but with low accuracy	P	10%	2,00E-05	0%	SM3	97,0%	6,00E-07	SM3	100%	0,00E+00
ADC	ADC	NSR			P	100%	7,00E-03				0,0E+00			0,0E+00
						Σ					0,00017			0,0E+00
						Total failure rate	0,0206							
						Total Safety Related	0,0136							
						Total Not Safety Related	0,0070							
										Single Point Faults Metric	98,8%			
												Latent Faults Metric		100%

[a] Depending on complexity it can be beneficial to have a dedicated entry in the FMEA giving more details about the potential root causes and the end effect of each failure mode.

[b] Fault model can be permanent fault (P) or transient fault (T); the example is limited to permanent faults.

5.3.6 Dependent failures analysis

As stated in ISO 26262-9:2011, 7.4.2, the analysis of dependent failures is performed on a qualitative basis because no general and sufficiently reliable method exists for quantifying such failures.

The steps reported in Clause 10 are applicable also for analogue and mixed signal components. In the dependent failures analysis, there are aspects that can be clearly considered when addressing analogue components, parts or sub-parts.

Analogue circuits are by nature sensitive to noise and interference among different blocks or functions. For this reason, structures to guarantee sufficient independence by means of isolation and separation (e.g. by implementing barriers and/or guard-rings or placing circuits at certain distances or separating the power supply distribution and even the ground layer) are implemented for functional reasons. In fact, substrate, power supply and global signals like bias, clock or reset are often considered as a source of interference and special care is taken to reduce such effect. This good design practice, usually followed for functional reasons, provides benefits in terms of dependent failure avoidance.

Analogue circuits can be very sensitive to process variation resulting in mismatches in the device behaviour. To ensure the "same" transfer function of two blocks, as in the case of redundant parts, the symmetry of the design and physical layout is a key factor. In such cases, special attention is taken to ensure exactly the same layout of the two blocks including orientation, symmetrical placing, routing etc.; therefore diversity is not always a viable solution to improve the common cause failure avoidance for analogue circuits.

As a consequence of these aspects, the dependent failure initiators are often addressed by techniques ensuring isolation or separation instead of with techniques aiming to differentiate their effects.

In other cases, diversity can still be a valid technique to achieve the detection or avoidance of dependent failures. For instance, in a dual channel approach, using two diverse ADC architectures (e.g. successive approximation ADC and sigma delta ADC) can reduce significantly the probability of common cause failures.

5.3.7 Verification of architectural metrics computation

This subclause is addressing a specific part of the safety analysis verification: the verification of the architectural safety metrics and in particular the fraction of safe faults and the failure mode coverage as defined in ISO 26262-10:2012, 8.1.

Possible approaches include:

— Expert judgment founded on an engineering approach given that any data, either qualitative or quantitative, is supported by rationale and relevant arguments included in the safety case.

NOTE 1 In some cases, such arguments can be derived from the functional characterization of the HW elements responsible for the claimed parameters. The aim of the functional characterization is the systematic failure avoidance and not the HW random failure but, in some cases, it can be used as evidence to prove the level of coverage with respect to a specific failure mode: This is the case in which the aim of a safety mechanism is to detect 100 % of one of more failure modes and this capability is guaranteed by design.

EXAMPLE 1 A voltage monitor is a typical safety mechanism used to detect overvoltage and under-voltage failure modes affecting the voltage regulator. If, during the HW design verification, the functional characterization of the voltage monitor shows that:

— any event leading to a regulated voltage outside the expected range defined in the specification for enough time to make the supplied HW circuit malfunction is detected by the voltage monitor; and

— any event leading to a variation of the regulated voltage inside the range defined in the specification for any time does not prevent the correct behaviour of the HW circuit supplied by the regulator;

then such characterizations can be used as arguments to claim a detection equal to 100 % of the mentioned failure modes.

— As mentioned in ISO 26262-5:2011, Table 11 and ISO 26262-10:2012, A.3.8.2, fault injection simulation during the development phase is a valid method to verify completeness and correctness of safety mechanism implementation with respect to hardware safety requirements and fault injection using design models can be successfully used to assist the verification. This method can be applied to analogue and mixed signal components.

NOTE 2 The fault injection campaign can be limited to a subset of faults or failures that are judged to be critical in a specific case. The most critical failure modes are identified taking into account their distribution, their claimed amount of safe faults, their claimed level of detection and the safety mechanisms or safety requirements responsible for those levels.

EXAMPLE 2 A failure mode is deemed too complex for expert judgement. This specific failure mode is a candidate to be characterized using fault injection.

— A combination of both methods, i.e. fault injection which supports expert judgment by providing arguments and evidence for the cases judged more critical and/or addressable by fault injection method alone.

5.4 Examples of safety mechanisms

The following tables give a non-exhaustive list of examples of commonly used analogue safety mechanisms and complement the information contained in ISO 26262-5:2011, Annex D.

Some analogue safety mechanisms have a digital output signal which is used to control the reaction to a failure and bring the component to a safe state. In many cases, this information is stored so that it can be communicated through a digital interface. Other analogue safety mechanisms control or suppress a fault from resulting in the violation of a safety requirement and do not interface with the digital domain.

To comply with ISO 26262-5:2011, 8.4.8, the safety mechanisms described in the following tables can require additional measures to detect faults affecting them that, as dual point faults, can lead to the violation of the safety goal.

The examples included in the following tables are not exhaustive and other techniques can be used.

Table 9 — Power supply

Safety mechanism/ measure	See overview of techniques	Notes
Over and under voltage monitoring	5.4.2	Typically an analogue circuit with an output latched in a digital core.
Voltage clamp (limiter)	5.4.3	Typically used to suppress voltage transients or spikes.
Over-current monitoring	5.4.4	Typically an analogue circuit with an output latched in digital core.
Current limiting	5.4.5	Typically an analogue circuit with feedback to an analogue control loop (e.g. to disable regulator main pass element).
Power on reset	5.4.6	Functional block which keeps the circuit in a known initialized state until power supply rails and/or the clock signal are stable.

Table 10 — Analogue I/O

Safety mechanism/ measure	See overview of techniques	Notes
Resistive pull up/down	5.4.1	Typically used on input signals to avoid floating conditions due to pin failure or external pin interconnect failure.
Filter	5.4.8	Analogue or digital circuit, typically used to suppress high frequency signal variation, like an output from analogue over and under voltage monitoring circuit.

Table 11 — Component

Safety mechanism/ measure	See overview of techniques	Notes
Analogue watchdog	5.4.7	
Thermal monitor	5.4.9	Typically an analogue circuit with an output latched in digital core, or feedback to an analogue circuit control loop (e.g. to disable affected circuit).
ADC monitoring	5.4.11	An analogue circuit typically controlled and evaluated by a digital circuit.
Analogue BIST	5.4.10	Typically an analogue circuit controlled by a digital circuit that verifies correct functionality of analogue safety mechanisms like under/over voltage monitoring, current limit protection and thermal protection circuits.

Table 12 — Analogue to digital converter

Safety mechanism/ measure	See overview of techniques	Notes
ADC attenuation detection	5.4.12	Typically an analogue circuit controlled by a digital circuit that validates the ADC conversion path by measuring a known and stable signal value.
Stuck on ADC channel detection	5.4.13	Typically an analogue circuit controlled by a digital circuit that validates the ADC conversion path by measuring a known and stable signal value.

5.4.1 Resistive pull up/down

Aim: To define a default voltage for a circuit node.

Description: A resistor is connected from a circuit node to either a supply voltage or ground to define a default voltage in the event that the driving signal becomes disconnected/high impedance. Commonly used on I/O pins.

EXAMPLE An un-driven or disconnected device/module input pin would be at an unknown voltage level. A pull-up resistor to the I/O supply voltage (or module supply voltage) or pull-down resistor to ground is used to keep the input at a known voltage level. The circuit itself could be a passive resistor or an active circuit like a current mirror.

5.4.2 Over and under voltage monitoring

Aim: To detect, as early as possible, when a regulated voltage is outside the specified range.

Description: The regulated voltage is compared via a differential input pair to a low and/or a high analogue reference voltage representing the limits of the specified operating range. The monitor output will change state when the regulated voltage is outside of the defined voltage window indicating a fault.

EXAMPLE A window comparator is used to monitor the output of a LDO regulator with reference voltages set to the minimum and maximum specified voltage levels in regulation.

5.4.3 Voltage clamp (limiter)

Aim: To prevent the voltage of a circuit node from exceeding the maximum voltage that can be safely supported.

Description: A voltage clamp limits the positive and/or negative voltage of a circuit node to an acceptable level determined by system and/or device process capability. Voltage clamps can be biased or unbiased. Unbiased clamps typically use Zener diodes to define the reference voltage while biased clamps use a voltage source in combination with specialized diodes (Zener, Schottky) to define the acceptable voltage level. Voltage clamps are typically used to protect against transient events.

EXAMPLE An ESD protection circuit is a specialized voltage clamp typically implemented on I/O pins. It is designed to shunt the energy of a high voltage electrostatic discharge on the I/O pins away from the internal circuitry to ensure that internal circuitry is not exposed to excessive voltage levels during the ESD event.

5.4.4 Over-current monitoring

Aim: To detect, as early as possible, when the output current exceeds a certain value.

Description: The implementation of over-current monitoring can vary. A typical approach for a voltage regulator circuit with an MOS output device is to add a sense FET in parallel with a regulator main FET. The sense FET current, which is proportional to the main FET current, flows across a sense resistor. The voltage drop across the sense resistor is amplified and monitored by a voltage monitor.

NOTE The output of an over-current monitor is a digital output which is subsequently used as feedback to an analogue circuit control loop, and/or latched in a digital core which interfaces to the control and/or status monitoring circuits.

5.4.5 Current limiter

Aim: To limit output current to a maximum level in order to maintain a safe operating area of the output device and prevent electrical overstress.

Description: A closed loop system using negative feedback from a current monitor to reduce the drive to the output device thereby limiting the output current.

5.4.6 Power on reset

Aim: To hold the outputs of a system in a known state (typically off) until internal nodes have stabilized upon power up or power reset conditions.

Description: Typically, a bandgap-derived voltage reference is compared to an attenuated supply voltage in order to detect the minimum specified supply voltage which will ensure correct operation. Hysteresis is typically required to prevent oscillation as the attenuated supply voltage exceeds the reference voltage.

EXAMPLE An under-voltage monitor is a mechanism used to detect and drive power-on reset.

5.4.7 Analogue watchdog

Aim: To monitor proper operation of an oscillator.

Description: Typically implemented with a monostable circuit (one shot) which is reset on each cycle of the oscillator. If an oscillator transition does not occur within a specified time period defined by the monostable circuit, a fault signal is produced.

5.4.8 Filter

Aim: A filter can be used in multiple ways as a safety mechanism and depends upon the safety requirement under consideration including:

EXAMPLE 1 A bypass capacitor can be used to suppress voltage transients. An RC time constant is used to qualify whether the duration of a fault which violates a safety requirement is within the fault tolerant time interval.

EXAMPLE 2 A digital de-glitch circuit can be used to filter level shifted analogue voltage comparator outputs. The de-glitch time duration is defined by the minimum signal transient duration that will be detected as a valid voltage fault condition.

5.4.9 Thermal monitor

Aim: To detect when circuit temperature exceeds a specified limit.

Description: Typically, a PTAT (proportional to absolute temperature) voltage is compared to a temperature independent reference voltage usually derived from a bandgap. The comparator will generate a fault signal when the PTAT voltage exceeds the reference voltage.

5.4.10 Analogue Built-in Self-Test (Analogue BIST)

Aim: Typically, to verify correct operation of diagnostic circuits and increase the detection of latent faults.

Description: The implementation of analogue BIST varies according to the diagnostic function to be verified. Analogue BIST typically involves exercising diagnostic circuits into and out of fault scenarios by injecting currents or voltages into the diagnostic circuit to ensure the diagnostic circuit can switch to both faulted and non-faulted states.

5.4.11 ADC monitoring

Aim: To measure an analogue signal by means of digital conversion with an output processed/evaluated in the digital core as an independent/ redundant analogue signal monitor.

Description: A critical analogue signal for which accuracy is relevant is converted in a digital code by means of an independent ADC (e.g. located outside the component or, at least biased by an independent source). The digital code is then processed by the CPU or an equivalent digital machine in order to determine if the original analogue signal has the required performance in terms of accuracy and static and dynamic behaviour. The frequency of the sampling and the resolution of the ADC and digital processing define which failure modes can be detected and to what accuracy.

5.4.12 ADC attenuation detection

Aim: To detect incorrect conversion of an analogue signal into its digital interpretation.

Description: Upon each background conversion loop, the element performs the conversion of the internal V_{mid} voltage both with and without the selectable attenuation switched in. The conversion results are stored respectively in separate SPI fields. A mathematical operation of dividing the attenuated result by the non-attenuated result verifies that the attenuation factor is within specified limits.

5.4.13 Stuck on ADC channel detection

Aim: To detect stuck on faults affecting the input signal to be converted by the ADC

Description: The element provides a multiplexer channel with series resistor RPOST, which is selected only when converting the test voltage channels (V_{high}, V_{low}, V_{mid}), and RPOST is otherwise bypassed. The value of RPOST is chosen such that a stuck-on channel within the post-buffer mux pulls one or more of the test voltage channels out of the expected voltage range.

EXAMPLE Each software loop, the MCU reads the ADC conversion results for the V_{high}, V_{low} and V_{mid} component ADC channels over SPI, and compares them against fixed detection thresholds.

5.5 About avoidance of systematic faults during the development phase

Analogue and mixed signal components are developed based on a standardized development process.

The general requirements and recommendations related to HW architecture and detailed design are defined in ISO 26262-5:2011, Clause 7.

The guideline in ISO 26262-10:2012, A.3.7 applies to the analogue and mixed signal components well if:

— ISO 26262-10:2012, Table A.8 is replaced by the following Table 13.

— ISO 26262-10:2012, A.3.7 f) is restricted to hard cores only

NOTE Wear and aging are considered during development with proper verification and validation procedures.

Table 13 — Examples of measures to avoid systematic failures in analogue and mixed signal components

ISO 26262-5:2011, Subclause	Design phase	Technique/Measure	Aim
6.5.1 HW safety requirements specification	**Specification**	Using an appropriate requirement management tool	To streamline the identification and tracking of the safety requirements for the HW element.
6.5.2 HW/SW interface specification		Using a model to describe HW/SW interface for critical elements	To reduce the risk of misinterpretation and to ensure consistency between HW and SW design.
7.5.1 HW design specification		Using an appropriate tool to allocate requirements to HW design	To streamline the identification and tracking of the design specification for the HW element.
7.4.1.6 Properties of modular HW design	**Design**	Use of modular, hierarchical, and simple design	The description of the circuit's functionality is structured in such a fashion that it is easily to understand. i.e. circuit function can be intuitively understood by its description without simulation efforts
7.4.1.6 Properties of modular HW design		HW design using schematics	Schematic entry is the method typically used for analogue circuitry.
7.4.4 Verification of HW design		Behavioural model simulation for critical elements	Behavioural models are simplified models of the design. Behavioural modelling for analogue circuits allows for the evaluation of functionality in an early design stage (e.g. to prove the design concept) and a reduction in simulation time.
7.4.4 Verification of HW design		Electrical model simulation	Simulation at transistor level is the method used to verify and validate the functionality of analogue circuitry.
7.4.4 Verification of HW design		Safe operating area (SOA) checks done by design review and/or tools	An analogue circuit is composed of devices with different current/voltage capabilities. SOA checking ensures that each device will work safely within its specific operational area according to its technology.

Table 13 *(continued)*

ISO 26262-5:2011, Subclause	Design phase	Technique/Measure	Aim
7.4.4 Verification of HW design		Corner simulations (i.e. technology process and environmental conditions spread)	In order to ensure block-level functionality, simulations are performed which take the spread of process parameters and environmental conditions into account.
7.4.4 Verification of HW design		Monte Carlo simulations of most sensitive blocks	In order to ensure block-level functionality of critical circuits, the effect of on-chip process spread is simulated using a statistical approach (i.e. Monte Carlo simulations)
7.4.4 Verification of HW design		Mixed mode simulations for critical elements	To ensure the correctness of critical elements, e.g. analogue to digital interfaces, analogue/digital closed loop control, digital circuits are simulated in the analogue domain.
7.4.4 Verification of HW design		Design for testability	Specific HW structures (e.g. test modes, multiplexers) are included into the design and layout in order to test otherwise inaccessible circuit nodes and improve the test coverage
7.4.4 Verification of HW design		Usage of coverage metrics to check the level of the verification	Verifies the completeness of the simulations and/or analysis by means of a quantitative approach (i.e. coverage metrics).
7.4.2.4 Robust design principles		Application of schematic design guidelines	Manual checks
7.4.4 Verification of HW design		Application of schematic checkers	Automatic tools
7.4.4 Verification of HW design		Documentation of simulation results	Documentation of all data needed for a successful simulation in order to verify the specified circuit function
7.4.4 Verification of HW design		Schematic design inspection or walk-through	Design review usually includes inspection or walk-through.
7.4.4 Verification of HW design		Application and validation of hard-core (reused schematic design and/or layout)	Usage of an already proven schematic or layout is highly recommended, especially for lower ASIL requirements.
7.4.4 Verification of HW design		Verification for behavioural models (if used) against the transistor level description	Cross check between behavioural model and the transistor level schematic design by simulation
7.4.4 Verification of HW design		Simulation of netlist with parasitics extracted from layout for critical elements	Back-annotated netlist simulated by analogue simulator
7.4.4 Verification of HW design		Verification of netlist with parasitics extracted from layout against the schematic netlist for critical elements	Back-annotated netlist is checked against the schematic description in terms of simulation results in order to take into account parasitic layout effects.
7.4.4 Verification of HW design		Layout inspection or walk-through (avoid cross talk between noisy and sensitive nets; avoid signal path with minimum width; use of multiple contacts/vias to connect layers)	The layout of analogue circuits is mainly done manually (automation is very limited with respect to the analogue blocks) and so layout inspection is crucial. The design review usually includes layout inspection or walk-through.

Table 13 *(continued)*

ISO 26262-5:2011, Subclause	Design phase	Technique/Measure	Aim
7.4.4 Verification of HW design		Design rule check (DRC)	The layout of analogue circuits is mainly done manually (automation is very limited with respect to the analogue blocks) and so design rule checking is more crucial than in the digital domain.
7.4.4 Verification of HW design		Layout versus schematic check (LVS)	The layout of analogue circuits is typically done manually (automation is very limited compared to the analogue blocks) and so checking layout versus schematic is more crucial than in the digital domain.
7.4.4 Verification of HW design	**HW design verification**	Development by HW prototyping	Verification of implemented functions by prototype (e.g. test chips, boards), can check particular points of the HW design where design review is not sufficient.
6.5.3 HW safety requirement verification report	**Verification**	HW safety requirement verification report	Provide evidence of consistency with HW specification, completeness and correctness
10.5.1 HW integration and testing activities	**HW integration testing**	Verification of the completeness and correctness of the design implementation on the component level	Perform component tests and reports
10.5.1 HW integration and testing activities		Usage of coverage metrics to check the level of testability	Verification of the completeness of the component tests
7.4.5 Production, operation, service and decommissioning 9.4.2.4 Dedicated measures	**Safety-related special Characteristics during Chip production**	Determination of the achievable test coverage of production test	Evaluation of the test coverage during production test with respect to the safety-related aspects of the component.
7.4.5 Production, operation, service and decommissioning 9.4.2.4 Dedicated measures		Determination of measures to detect and cull early failures	Assurance of the robustness of the manufactured component. In most, but not every process, gate oxide integrity (GOI) is the key early life failure mechanism. There are multiple methods of screening early life GOI failures including high temp/high voltage operation (Burn-In), high current operation and voltage stress however these methods could have no benefit if GOI is not the primary contributor to early life failures in a process.
7.4.5 Production, operation, service and decommissioning 10 Hardware integration and testing	**Qualification of HW component**	Definition and execution of qualification tests like Brown-out test, High Temperature Operating Lifetime (HTOL) test and functional test-cases, Specification of requirements related to production, operation, service and decommission HW integration and testing report	For an analogue component with integrated brown-out detection, the component functionality is tested to verify that the outputs of the analogue circuit are set to a defined state (for example by stopping the operation of the analogue circuits in the reset state) or that the brown-out condition is signalled in another way (for example by raising a safe-state signal) when any of the supply voltages monitored by the brown-out detection reach a low boundary as defined for correct operation. For an analogue component without integrated brown-out detection, the analogue functionality is tested to verify if the analogue circuit sets its outputs to a defined state (for example by stopping the operation of the analogue circuit in the reset state) when the supply voltages drop from nominal value to zero. Otherwise an assumption of use is defined and an external measure is considered.

5.6 About safety documentation

Analogue and mixed-signal components are predominantly developed within a distributed development due to the specific nature of their functionality.

Guidelines reported in ISO 26262-10:2012, A.3.10 for SEooC components can be used as a reference for the safety work products to be exchanged, however, an adaptation to the different development approach can be necessary. For example, in the case of an SEooC component, the safety analysis is usually performed in a worst case condition while for analogue components the analysis is more oriented to the target context. Joint expertise from both the end user and the supplier is therefore important.

— The DIA between the component manufacturer and the end user specifies which documents are to be made available from each party as well as the level of work-share between the parties.

— The safety requirement specification defines the expected functionality of the component. It is critical that such specifications are carefully compiled by the end user, according to ISO 26262-8:2011, Clause 6, to ensure that correct functionality is understood by all suppliers in the distributed development. A description about the usage of the elements of the component as well as identification of pre-defined on-chip/off-chip safety mechanisms is important to allow a proper safety analysis at a system level (e.g. to allow fault classification into safe, potential to violate a safety goal, etc., for each safety goal considered).

NOTE 1 If the component is developed out of context, the requirements derived from the technical safety concept are replaced by assumptions of use.

Documentation describing the capabilities of analogue and mixed signal components are listed below:

— The results of the checks against the applicable requirements of ISO 26262 including confirmation measures reports (if applicable);

— Safety analysis results as per agreement; (These can be simply raw failures of the component, their distribution and diagnostic coverage offered from the specified safety mechanisms or a full FMEA for different safety requirements.)

— Information regarding the calculation of the failure rate (e.g. number of transistors);

— A description of any assumptions of use of the component with respect to its intended usage.

NOTE 2 Such documentation can be combined into one document constituting a “Safety Manual” or “Safety Application Note” of the analogue or mixed signal component

6 Intellectual property and ISO 26262

6.1 About intellectual property

6.1.1 Understanding intellectual property

Intellectual property (IP) refers to a reusable unit of logical design or physical design intended to be integrated into a design as a part or a component. The term IP integrator is used in this document for the organization responsible for integrating intellectual property designs from one or more sources into a design with safety requirements. The term IP supplier is used in reference to the organization responsible for developing or having developed the IP design. An IP design can be provided by the intellectual property integrator or a separate party, possibly in a different organization or company.

In a product development project involving intellectual property, the allocation of responsibilities between the IP supplier and the IP integrator can vary depending on the project. This division of responsibility requires effective safety management in terms of safety planning, and agreement on the development interfaces. For these activities the requirements of ISO 26262-2:2011, Clause 6 and ISO 26262-8:2011, Clause 5 are applied.

Based on the requirements in ISO 26262 four possible approaches are identified for IP based designs. These approaches are shown in Figure 5 with references to appropriate clauses within this document. The IP integrator typically chooses the approach based on consideration of the information provided from the IP supplier as well as the maturity of the IP.

EXAMPLE If no supporting information is available from the IP supplier, the possible approaches can be limited to hardware qualification based argument or proven in use argument.

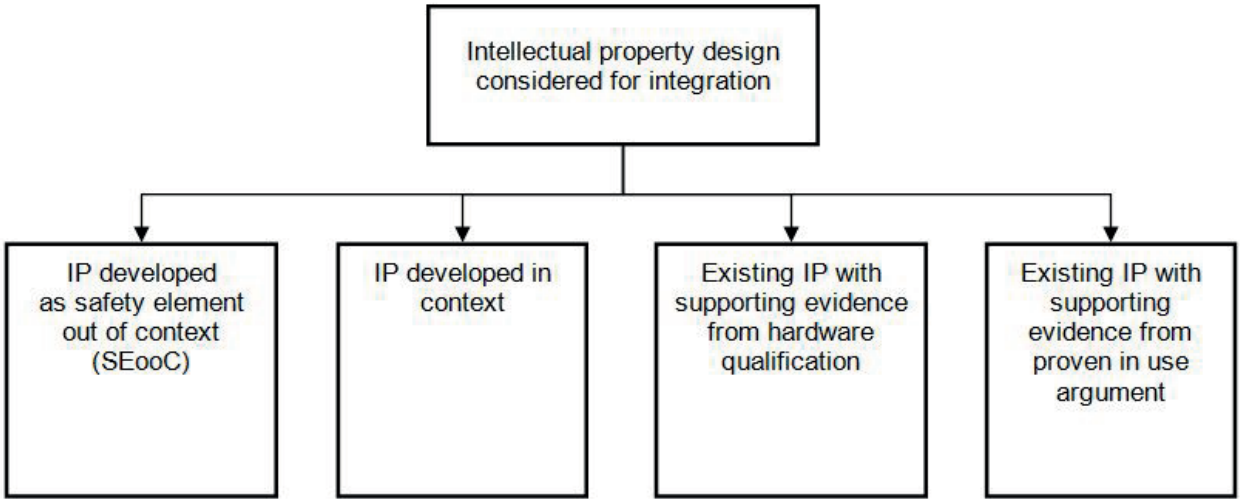

Figure 5 — Four possible approaches for using IP in safety-related designs

The intellectual property can be an existing design with a pre-defined set of features. In this case the IP integrator has the responsibility of identifying the set of features which are required to support the safety concept of the design. Intellectual property can also be designed based on an agreed set of safety requirements. In this case the IP integrator identifies the requirements for the IP which are necessary to support the safety concept of the design. These IP use models are further described in section 6.

The guidance in this document can be applied to newly developed IP, modified IP, and existing unmodified IP. However, the requirements in ISO 26262-8:2011, Clause 13 and Clause 14 can restrict the applicability of parts of this document to existing IP only.

Concerning the development of IP, a common approach is to assume the possible target usage as defined in ISO 26262-2:2011, 6.4.5.6. This option is described as safety element out of context (SEooC) in ISO 26262-10:2012, Clause 9. Development of an SEooC relies on identification of assumed uses and safety requirements which shall be verified by the IP user.

6.1.2 Types of intellectual property

Commonly used intellectual property types are listed in Table 14. This is not an exhaustive list covering the possible intellectual property types. This document considers both types of intellectual property as applied to semiconductor designs.

Table 14 — Types of intellectual property

Intellectual property type	Description
Physical representation	A complete chip layout description, containing instantiations of standard cells for a specific cell library for a target manufacturing process. EXAMPLE A/D converter macro, PLL macro
Model representation	A description of a design in terms of a hardware description language (HDL) such as Verilog or VHDL, or analogue transistor level circuit schematic. A logic design in model representation needs to be synthesized into a list of gates consisting of basic cells, followed by placement and routing to achieve a semiconductor design. Analogue circuit schematic components, such as transistors, diodes, resistors, and capacitors, need to be mapped into target technology library components, followed by placement and routing to achieve a semiconductor design. EXAMPLE Processor or memory controller design exchanged without mapping to a particular technology, operational amplifier transistor level schematic.
NOTE 1 IP in the form of chip layout is also known as hard IP. NOTE 2 IP in the form of logic design as soft IP. NOTE 3 In addition to digital logic, the guidance in this document is also applicable to analogue IP designs.	

Intellectual property in the form of logic design can also be configurable. The configuration options are typically specified by the IP integrator at the time of logic synthesis.

EXAMPLE 1 Configuration options to define interface bus width, memory size, and presence of fault detection mechanisms.

Intellectual property can also be generated with dedicated tools. Since the tool output will determine the functionality, sufficient confidence in software tools can be demonstrated using the methods described in ISO 26262-8:2011, Clause 11.

EXAMPLE 2 Memory compilers, C to HDL compilers, Network-on-Chip generators.

6.2 Safety requirements for intellectual property

In general, two categories of intellectual property can be determined based on the allocation of safety requirements: IP with no allocated safety requirements, and IP with one or more allocated safety requirements. When the intellectual property has no allocated safety requirements, QM development is applicable and no additional considerations are required for ISO 26262. For designs incorporating IP with QM requirements, dependent failure analysis can be used to demonstrate freedom from interference of the integrated IP with other safety-related design elements. For dependent failure analysis guidance, see ISO 26262-9:2011, Clause 7.

If the intellectual property is allocated one or more safety requirements, ISO 26262 requirements are applicable. In particular, ISO 26262-2, ISO 26262-5, ISO 26262-8, and ISO 26262-9 contain applicable requirements for IP designs. The following text gives guidance on IP with allocated safety requirements, and how to consider these requirements for IP with and without integrated safety mechanisms.

IP with one or more allocated safety requirements can be further classified based on the integration of safety mechanisms. Two possible cases are illustrated in Figure 6, with subfigure (a) illustrating IP which has safety mechanisms, and subfigure (b) illustrating IP which has no integrated safety mechanisms.

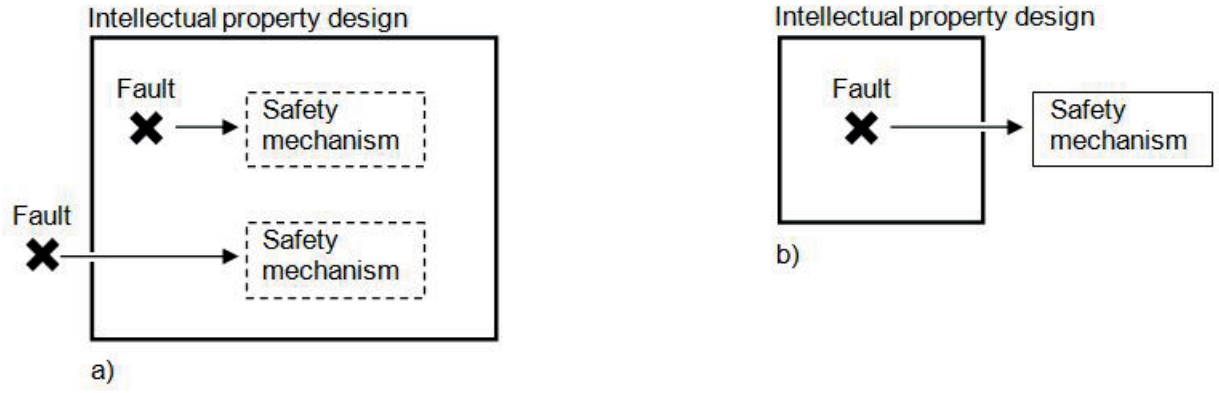

Figure 6 — Types of IP with allocated safety requirements

NOTE 1 IP safety mechanisms can be included for detection of faults internal to the IP, as well as faults external to the IP.

NOTE 2 Safety mechanisms implemented in the IP can also provide a partial coverage of a defined set of faults. It is also possible that only fault detection is performed by the IP, with fault control being provided by components external to the IP.

The IP integrator can decide to allocate safety requirements to specific hardware features of the IP. The IP integrator is responsible for determining the suitability of the IP hardware features to satisfy the technical safety requirements and hardware safety requirements.

The hardware features of the IP can be initially developed targeting its integration into a safety-related hardware environment, by providing safety mechanisms based on assumed safety requirements that aim at controlling given failure modes. In this case the requirements of ISO 26262-2, ISO 26262-5, ISO 26262-8, and ISO 26262-9, whenever applicable, can be used for the design of the safety mechanisms during the development of the IP.

EXAMPLE 1 Bus fabric with built-in bus supervisors including error reporting logic (e.g. interrupt signals) and diagnostics (error capture information).

EXAMPLE 2 Voltage regulator with monitoring (under-voltage and over-voltage detection), protection (current limit or thermal protection) and diagnostics (monitoring and protection circuit built-in self-tests).

Alternatively the IP can be developed with no assumed safety requirements or specific safety mechanisms to detect and control faults.

EXAMPLE 3 Bus fabric without built-in bus supervisors or error reporting logic. Voltage regulator without monitoring, protection or built-in monitoring or protection circuit diagnostics.

EXAMPLE 4 Voltage regulator without monitoring, protection or built-in monitoring or protection circuit diagnostics.

For IP with safety mechanisms, safety analyses defined in ISO 26262-9:2011, Clause 7 and Clause 8 are applicable. A qualitative safety analysis can be provided to the IP integrator to justify the capabilities of the safety mechanisms to control given failure modes. Similarly a dependent failure analysis can be provided to demonstrate required independence or freedom from interference.

NOTE 3 The IP supplier includes example information concerning failure mode distribution in the safety analysis results, based on specific implementation assumptions. Documentation related to safety mechanisms can be provided with other safety-related documentation for the IP. This information can also be combined into a single safety manual or safety application note as described in ISO 26262-10:2012, A.3.10.

NOTE 4 This information can be contained within existing documentation (e.g. integration guidelines, technical reference documents, application notes).

The IP integrator can request additional information from the IP supplier in implementing safety requirements. The IP supplier can support the request by providing information concerning measures used to avoid systematic faults, as well as safety analysis results. Safety analysis results can be used to support the determination of hardware metrics for the integrated IP, as well as to demonstrate freedom from interference and independence.

Since the IP will be integrated into a safety-related design, consideration of freedom from interference is important to ensure that the integrated IP can have no adverse impact on other safety-related functions. For the freedom from interference and independence claims, dependent failure analysis as described in ISO 26262-9:2011, Clause 7 can be used, together with the additional guidance in Clause 10 of this document.

EXAMPLE 5 The IP integrator decides that a JTAG debug IP for a processor core without safety mechanisms can be integrated without modification as it has no adverse effects on the integrated design.

If the IP integrator determines that the fulfilment of safety requirements is not possible with the supplied IP, a change request to the supplier can be done as described in ISO 26262-8:2011, 5.2 and ISO 26262-10:2012, 9.2.2 in cases where the IP is an SEooC. Alternatively, other measures by the IP integrator to satisfy safety requirements can be applied, including integration of logic to allow the use of ASIL decomposition as described in ISO 26262-9:2011, Clause 5 and related example in ISO 26262-10:2012, Clause 11, or additional safety mechanisms at integration level. Additional safety mechanisms can be implemented in hardware, software, or combination of both.

The IP integrator is responsible for all integration and associated verification and testing activities related to the allocated safety requirements and safety mechanisms, as applicable.

NOTE 5 The IP supplier provides supporting information to allow the IP integrator to conduct integration activities, including information on the verification and testing done for the IP.

6.3 Intellectual property lifecycle

6.3.1 ISO 26262 and the intellectual property lifecycle

Avoidance and detection of systematic faults during the intellectual property lifecycle is required to ensure that the resulting design is suitable for use in applications with one or more allocated safety requirements. Requirements for avoidance and detection of systematic faults are provided in ISO 26262-5. ISO 26262-10:2012, Table A.8 provides further guidance for microcontroller designs. This table can be used to determine the general methods that can be used during IP development to avoid and detect systematic faults. Due to the wide range of IP designs with differing functionality and complexity, guidelines from ISO 26262-10:2012, Table A.8 need to be appropriately interpreted.

For IP which exhibits programmable behaviour, ISO 26262-4:2011, 7.4.5.2 can be considered.

The IP integrator is responsible for integrating the supplied IP. For the integration activities the assumptions of use and integration guidelines described for the IP are considered. The impact of assumptions of use which cannot be fulfilled or that are invalid with the design into which the IP is being integrated is analysed and considered with change management conducted as described in ISO 26262-8:2011, Clause 8. Figure 7 illustrates the flow following the guidance already provided in ISO 26262-10:2012, 9.2.3.1.

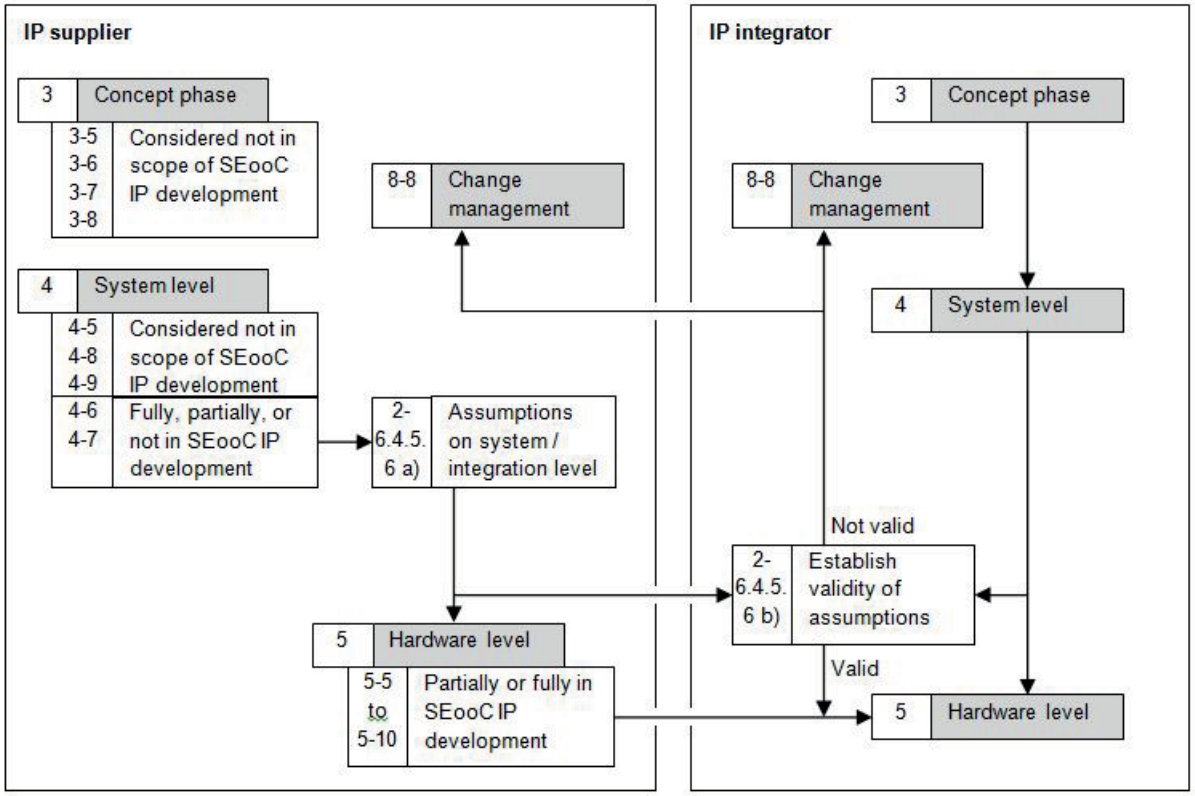

Figure 7 — IP lifecycle when IP is treated as SEooC

NOTE ISO 26262-5, Clause 10 is shown to be the responsibility of the IP supplier. However, this clause has a number of requirements which are not applicable to IP suppliers.

In order to support the IP integrator in IP integration activities, the IP supplier can provide information about the IP in the form of defined work products. The work products allow the IP integrator to determine applicable requirements for the supplied IP. Additional information can be provided to support safety analysis activities.

6.3.2 Intellectual property as safety element out of context (SEooC)

When developing an SEooC IP, applicable safety activities are tailored as described in ISO 26262-2:2011, 6.4.5.6. Such tailoring for the SEooC development does not imply that any step of the safety lifecycle can be omitted. In case certain steps are deferred during the SEooC development, they are completed during the item development.

The ASIL capability of an SEooC designates the capability of the SEooC to comply with assumed safety requirements assigned with a given ASIL. Consequently, it defines the requirements of ISO 26262 that are applied for the development of this SEooC.

In cases where a mismatch exists between the SEooC ASIL capability and the ASIL requirements specified by the IP integrator, the IP integrator can implement additional safety mechanisms external to the IP. Additional safety measures for systematic failure avoidance are also considered. It is possible to use ASIL decomposition as defined in ISO 26262-9:2011, Clause 5, provided that the methods for systematic failure avoidance and control for the integrated IP are taken into account.

An SEooC is therefore developed based on assumptions of the intended functionality and use context which includes external interfaces. These assumptions are set up in a way that addresses a superset of components integrating the SEooC, so that the SEooC can be used later in multiple different designs. The validity of these assumptions is established in the context of the actual component integrating the SEooC.

IP developed as an SEooC can often be configured to target a number of different designs. Configuration can be done before synthesis, after synthesis, or by programming. Information provided by the IP supplier can include information on the IP configurations which have been covered by testing and verification activities.

EXAMPLE Configuration options to determine bus width for interconnects, internal cache memory sizes, number of interrupts, memory maps.

NOTE IP configuration differs from configuration data for software, as described in ISO 26262-6:2011, Annex C.

6.3.3 Intellectual property designed in context

When developing IP in context, the IP supplier tailors the safety activities as described in ISO 26262-2:2011, 6.4.5.1. For in context designs, the IP supplier can develop the IP with knowledge of the safety requirements. Similarly safety analyses for the IP can use all available information included in the item definition.

The documentation for IP designed in context does not differ significantly from documentation for SEooC IP.

6.3.4 Intellectual property use through hardware component qualification

In cases where no SEooC or in-context information is available for the IP, hardware qualification as described in ISO 26262-8:2011, Clause 13 can be used to increase confidence in the IP. Hardware qualification activities can be applied to IP without supporting information.

6.3.5 Intellectual property use through proven in use argument

Since IP tends to be widely re-used, proven in use argument as described in ISO 26262-8:2011, Clause 14 can provide a means for the IP integrator to demonstrate that an IP design is appropriate for a particular application.

The conditions surrounding the validity of the proven in use argument can be restricting. In particular, ensuring that an effective field monitoring program described in ISO 26262-8:2011, 14.4.5.3 is in place can be challenging due to the typically limited field feedback from designs incorporating IP.

6.4 Work products for intellectual property

6.4.1 ISO 26262 and work products for intellectual property

ISO 26262-10:2012, A.3.10 describes example work products for an SEooC microcontroller. This section contains guidance on contents of work products which can be provided for IP designs in general.

NOTE A development interface agreement (DIA) can be used to specify which documents are made available to the IP integrator and what level of detail is included.

6.4.2 Safety plan

For IP with one or more allocated safety requirements, the safety plan is developed based on the requirements in ISO 26262-2:2011, 6.4.3.5. A single plan or multiple related plans can be used. Detailed plans are included for applicable supporting processes as described in ISO 26262-8, covering configuration management, change management, impact analysis and change requests, verification, documentation management and software tool qualification.

6.4.3 Safety requirements and verification review of the IP design

The safety requirements can be allocated to the IP design as defined in ISO 26262-5:2011, Clause 6.

EXAMPLE The requirement for a safety mechanism in the IP is described, allowing the requirement to be verified at appropriate level of integration. The integration and test requirements can be linked to requirements defined in the technical safety concept.

Defining criteria for design verification, in particular for environmental conditions (temperature, vibration, EMI, etc.) for an IP design which is provided in the form of logic design is not typically possible since the physical characteristics are highly dependent on the physical implementation of the design by the IP integrator.

A verification report includes results of the activities used to verify the IP design. Verification can be done as described in ISO 26262-8:2011, Clause 9, including planning, execution, and evaluation of verification activities.

NOTE Fault injection can be done using simulation as described in ISO 26262-5:2011, 7.4.4.1. Further guidance on fault injection for complex IP designs is included given in ISO 26262-10:2012, A.3.8.2.

6.4.4 Safety analysis report

The requirements for safety analysis in ISO 26262-9:2011, 8.4 are applicable for IP designs. The selection of appropriate safety analysis methods is based on ISO 26262-5:2011, Table 2.

For qualitative analysis, to support the integration of the IP the IP supplier can provide information about identified failure modes for the IP.

For quantitative analysis, the data included supports the evaluation of hardware architectural metrics and evaluation of safety goal violations due to random hardware faults, as specified in ISO 26262-9:2011, 8.4.10.

EXAMPLE Data includes estimated failure rate and failure mode distribution information.

NOTE 1 For IP provided as logical design, such as RTL, quantitative analysis relies on assumptions about failure rates and failure mode distributions, and can therefore not be representative of actual physical designs. The IP integrator verifies the assumptions and quantitative safety analysis results for the specific implementation.

NOTE 2 In estimating the metrics, safety mechanisms embedded in the IP and their expected failure mode coverage at a level that is applicable to given IP can be taken into account.

In the case of configurable IP, the safety analyses can include information about the impact of configuration options on the failure modes distribution.

Additional safety mechanisms realized by a combination of features internal and external to the IP, as well as safety mechanisms implemented outside the IP can be described. These additional safety mechanisms can rely on assumptions of use for SEooC designs, which can be validated at the appropriate level as described in ISO 26262-2:2011, 6.4.5.6.

6.4.5 Analysis of dependent failures

Dependent failure analysis for IP can be performed as described in ISO 26262-9:2011, Clause 7. Additional guidance on how to apply dependent failure analysis for semiconductor devices is included in Clause 10 of this document.

6.4.6 Confirmation measure reports

Reports from conducted confirmation measures include evidence and arguments related to the IP development process and about avoidance of systematic faults. Confirmation measures are described in ISO 26262-2:2011, Table 1. For semiconductor IP typical confirmation measure reports include:

— Confirmation review of the safety plan;

— Confirmation review of the safety analyses;

— Confirmation review of the software tool criteria evaluation report;

— Confirmation review of the proven in use arguments, if applicable;

— Confirmation review of the completeness of the safety case;

— Functional safety audit and assessment reports.

Examples of techniques applicable to IP development activities for systematic fault avoidance are included in ISO 26262-10:2012, A.3.7 and Table A.8.

6.4.7 Development interface agreement

The requirements for development interface in ISO 26262-8:2011, Clause 5 can be applied to IP designs. A development interface agreement defines the exchanged work products for IP designs, and the roles and responsibilities for safety between the IP supplier and the IP integrator.

6.4.8 Integration documentation set

An integration documentation set can include a safety manual or safety application note for IP developed as an SEooC. The integration documentation set can also include the following information:

— Description of the tailoring of the ISO 26262 lifecycle for the IP development.

— Description of the safety architecture, including

  — Fault detection and control mechanisms.

  — Fault reporting capabilities.

  — Self-test capabilities and additional requirements for self-testing for potential latent faults, if applicable.

  — Fault recovery mechanisms, if applicable.

— Assumptions of use for the IP, including for example:

  — Assumed safe states of the IP.

  — Assumptions on fault tolerant time interval and multiple-point fault detection interval.

  — Assumptions on the integration environment for the IP, including interfaces.

— Software configuration required to support the IP safety mechanisms, and to control failures after detection.

— Description of safety analysis results for the IP.

— Description of confirmation measures used for the IP.

NOTE 1 The above information can be included in one or more separate documents.

It is possible for the IP integrator to formally identify all the hardware properties related to the safety mechanisms so that a mapping with technical safety requirements and hardware safety requirements at the level of the IP integrator can be done, and the verification and validation activities that are the responsibility of the IP integrator can be identified.

NOTE 2 The IP safety mechanism requirements are specified in a way which allows them to be traceable to IP integrator's requirements.

NOTE 3 For IP with no specific features for fault detection, providing the assumptions of use can be sufficient to meet the IP integrator's requirements.

For IP developed in-context, similar documentation is typically provided. For in-context IP, assumptions of use are not required, as the IP is designed with full context information in place.

6.5 Integration of black-box intellectual property

In some projects the IP integrator can encounter a situation where it is necessary to integrate an IP in which contents are not fully disclosed. The IP to be integrated is a "black box" from the perspective of the IP integrator.

Reasons why a black box IP can be integrated include:

— IP integrator's customer requires use of their proprietary logic, such as a specific communications interface, timer peripheral, or similar logic;

— IP integrator is asked to integrate logic from a competitor, in order to facilitate a multi-source supply agreement.

Black box IP can be integrated in many forms, including but not limited to:

— Pre-hardened, or handed off as a gate level layout;

— As encrypted netlist, which cannot be meaningfully parsed except by trusted tools;

— As obfuscated RTL source (where meaningful variable names are replaced with randomized character strings and any explanatory comments are removed).

NOTE 1 A black box integration approach can also be applied to cases in which no information is available from the IP supplier.

When black box IP is integrated, the responsibility between IP supplier, IP integrator and the IP integrator's customer can be defined through a development interface agreement as described in ISO 26262-8:2011, Clause 5.

EXAMPLE 1 In cases where the IP integrator is required to use black box IP, for example because of a requirement from their customer, the DIA can specify that it is the customer responsibility to evaluate and accept the suitability for the use of the black box IP in a safety-related context.

The development interface agreement can also include details about the tailoring of the safety activities as described in ISO 26262-2:2011, 6.4.5.6 and the exchange of documentation across the supply chain.

EXAMPLE 2 A development interface agreement can specify that integration details are provided by the IP supplier in the form of an integration guide, also containing a set of validation tests which can be used to confirm proper integration.

Unless the IP has been developed specifically targeting the automotive market it is possible that ISO 26262 specific evidence is not available. In this case the responsibility for the acceptance available evidence can be defined in the development interface agreement.

EXAMPLE 3 IP developed according to other functional safety standards such as IEC 61508:2010.

NOTE 2 In this case information on the development lifecycle and associated processes used to develop the IP can be used to perform a gap analysis to evaluate the suitability of the IP for use in an ISO 26262 context.

The IP integrator does not always have enough data to evaluate the base failure rate of a black box IP. Since this can affect the results of quantitative analysis, the development interface agreement can specify the responsibilities between the IP supplier, IP integrator and the IP integrator's customer for

the estimation of the base failure rate. The responsibilities for safety analysis of the black box IP can be defined in a similar way.

NOTE 3 The integration of black box IP into a hardware development has parallels in software development, such as the case in which a developer integrates unit software from a third-party supplier as compiled object code. As such, the integrator of black box IP into a hardware development can find methods and techniques from the software domain and specifically ISO 26262-6 helpful in defining their integration strategy.

It is possible that the black box IP does not include diagnostics mechanisms. The ability of the IP integrator to add diagnostics can be limited due to the limited information about the black box IP. As a result, redundancy based techniques such as lockstep compare and 1oo2 voters can be used to provide diagnostic capability for black box IP. In addition, state estimation logic as a diagnostic monitor can also be used if enough data can be obtained via testing on the behaviour of outputs for known inputs.

7 Multi-core components and ISO 26262

7.1 Types of MC components

The following table summarizes the different types of MC components considered in this document.

Table 15 — Types of MC components

MC component type	Description
Homogeneous MC component	Homogeneous MC components include only identical PE
Heterogeneous MC component	Heterogeneous MC components have non-identical PEs, typically with different ISAs

EXAMPLE Figure 8 shows a diagram of a generic homogeneous dual-core system, with CPU-local level 1 caches, and a shared, on-die level 2 cache.

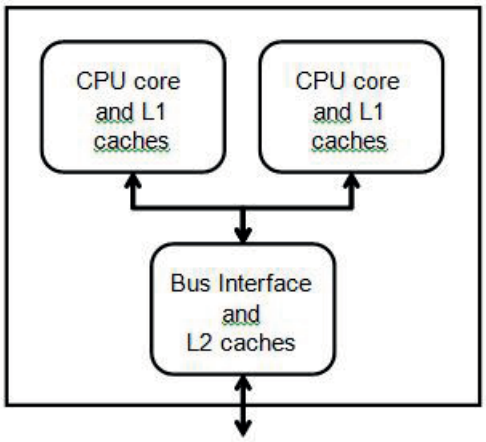

Figure 8 — Generic diagram of a dual-core system

7.2 Implications of ISO 26262 on MC components

7.2.1 Introduction

This clause addresses guidance for semiconductor vendors as also for system developers for which safety requirements – previously allocated to multiple components – are now allocated to a multi-core.

In particular, this clause addresses the following MC related topics and the respective implications of ISO 26262:

— ASIL decomposition in MC components;

— coexistence of elements with different ASILs in MC components;

— freedom from interference in MC components;

— dependent failure analysis in MC components;

— impact of timing requirements in MC components and the related implications.

7.2.2 ASIL decomposition in MC components

As shown in Figure 9, the initial safety requirement can be decomposed to two (or more) safety requirements which are allocated to sufficiently independent hardware and/or software elements.

EXAMPLE An ASIL B safety requirement is decomposed in two redundant requirements, ASIL B(B) – satisfied by the software running in PE1 - and QM(B), satisfied by the software running in PE2.

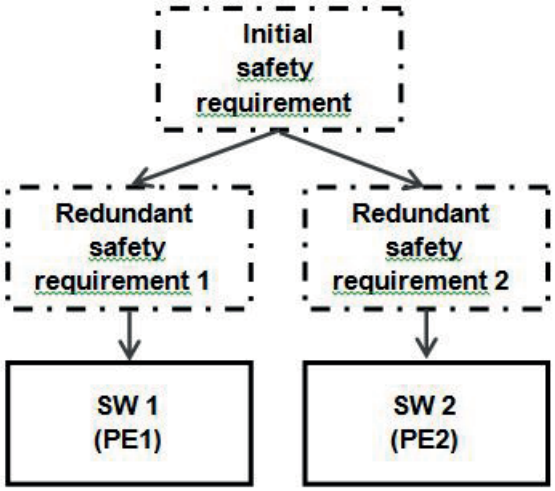

Figure 9 — ASIL decomposition in the context of MC

NOTE 1 In case that a software or hardware based comparator is used to compare the results of SW 1 and SW 2, the comparator will be developed according the original ASIL.

NOTE 2 SW redundancy can be implemented also outside the ASIL decomposition, as a safety mechanism to provide diagnostic coverage.

EXAMPLE Examples of SW redundancies are: SW heterogeneous redundancy; SW architecture in which a redundant copy of the software is executed by two identical PE in parallel and then compared by another SW unit. This technique is usually referred to as software lock-step or loosely coupled lock-step. A description of those types of SW redundancies is not in the scope of this clause.

This decomposition follows the requirements described in ISO 26262-9:2011, Clause 5. Guidelines of application to MC components are given in the following sub-clauses.

NOTE 3 ASIL decomposition has effect on both HW and SW systematic failures. This section provides clarifications only with respect to the SW level, for example how shared resources are considered in that context. Moreover, this section provides clarification on how requirements on the evaluation of the hardware architectural metrics and the evaluation of safety goal violations due to random hardware failures of the MC component remains unchanged by ASIL decomposition. It also clarifies how the SW redundancy inherently related to ASIL decomposition can be considered in the metrics evaluation.

7.2.2.1 Requirements decomposition with respect to ASIL tailoring on SW level

Application of ASIL decomposition between two or more diverse software elements is possible if sufficient independence regarding SW caused dependent failures can be shown between the corresponding SW elements.

Shared resources are a known DFI. For a SW element a shared resource can be a hardware element (e.g. cores, RAM, cache) as well as a software element (e.g. drivers). Within a MC the issue of shared cores (e.g. memory, time, execution or exchange of information interferences) can be resolved by assigning the corresponding SW elements to different PEs. Other issues (e.g. shared memory, commonly used SW elements) are addressed analogue to a single core system (e.g. memory encapsulation via MPU by the OS, developing the commonly used SW elements compliant with the initial ASIL).

NOTE Safety mechanisms ensuring the independence of the corresponding SW elements are implemented compliant with the initial ASIL and not with the decomposed ASIL.

EXAMPLE 1 The task to read and monitor an external sensor is allocated to the SW. The initial requirement is rated with an ASIL X. In the further development steps this requirement is allocated to SW element SW_Mon.1 with an ASIL Y(X) and to software element SW_Mon.2 with an ASIL Z(X). A DFA has shown that next to other issues the shared resources (cores, RAM and a SW driver “SW_Peripheral” forwarding the sensor values to SW_Mon.1 and SW_Mon.2) can threaten the independence requirement, i.e. causing memory, time, execution or exchange of information interferences between SW_mon.1 and SW_mon.2. In this example the shared core issue is addressed by mapping SW_Mon.1 and SW_Mon.2 to two different PEs, therefore unsharing the cores. The memory interference aspect is addressed by memory encapsulation via a MPU which is configured by the OS. Since in this case the OS is a safety mechanism ensuring the independence between SW_Mon.1 and SW_Mon.2 it is developed compliant with ASIL X. The issue with the shared SW resource “SW_Peripheral” is addressed by developing it compliant with the initial ASIL, ASIL X.

When applied, ASIL decomposition requires a sufficient level of independence between the redundant elements. As stated in ISO 26262-9;2011, 5.4.11 Note b), “sufficient” does not mean completely independent. Sufficient independence can be achieved not only by prevention of dependent failures but also by detection and mitigation of dependent failures at appropriate levels depending on allocated safety requirements.

Dependent failure analysis (DFA) as described in 7.2.6 is conducted to ensure that sufficient independence exists between the elements implementing the decomposed requirements. If sufficient independence cannot be shown, appropriate measures are applied at the software, hardware, and/or system levels to achieve sufficient independence.

Dependent failures can also be mitigated by using independent hardware units in combination with different software and/or dedicated software protocols.

EXAMPLE 2 The cross-check between two PEs is implemented by means of a dedicated software protocol running on a third independent and different PE.

7.2.2.2 Application of ISO 26262-9:2011, 5.4.5 to MC components

The requirements on the evaluation of the hardware architectural metrics and the evaluation of safety goal violations due to random hardware failures of the MC component remain unchanged by ASIL decomposition in accordance with ISO 26262-5.

An ASIL decomposition by itself has no impact on the metric evaluation, i.e. no metric requirements are altered as a result of ASIL decomposition.

EXAMPLE 1 As described in the example of ISO 26262–9:2011, 5.4.9, an ASIL D requirement is first decomposed into one ASIL C(D) requirement and one ASIL A(D) requirement. Then the ASIL C(D) requirement can then subsequently be decomposed into one ASIL B(D) requirement and one ASIL A(D) requirement, each mapped to a different PE. The decomposition has no impact on the necessity to evaluate the HW metrics compliant with ASIL D requirements of the item, i.e. the ASIL decomposition procedure does not automatically infers a lower ASIL requirement as far as the metrics evaluation is concerned: a safety analysis is needed to verify the overall metric compliance to the initial ASIL requirement.

Because the requirements on the evaluation of safety goal violations due to random hardware failures of the MC component remain unchanged by ASIL decomposition, the normative requirements for ASIL C and ASIL D as given in ISO 26262-5:2011, Clause 9 are applicable, including:

— 9.4.2.5, 9.4.3.10, and 9.4.3.11 (for ASIL C and ASIL D) on item-level;

— 9.4.3.8 (for ASIL D);

— 9.4.3.9 (for ASIL C)

EXAMPLE 2 In the case of an ASIL D decomposition into ASIL B(D) for PE1 and ASIL B(D) for PE2, both PE1 and PE2 will be considered as driven by ASIL D requirement. For example, according to ISO 26262-5:2011, 9.4.3.8, a dual-point failure is considered plausible if one or both hardware parts involved have a diagnostic coverage (with respect to the latent faults) of less than 90 %; or one of the dual-point faults causing the dual-point failure remains latent for a time longer than the multiple-point fault detection interval as specified in requirement ISO 26262-5:2011, 6.4.8. ISO 26262-10:2012, Figure A.1 gives a proposal how to define part in this context.

The fact that the requirements on the evaluation of the hardware architectural metrics and the evaluation of safety goal violations due to random hardware failures of the MC component remain unchanged by ASIL decomposition does not mean that the SW redundancy inherently related to ASIL decomposition does not bring any contribution to the metrics computation.

EXAMPLE 3 With reference to Figure 9, in the metrics computation and in particular in the evaluation of diagnostic coverage of PE1 and PE2, it is still possible to take credit of the fact that a final comparison will occur between SW1 and SW2. However, in general, that comparison is not able to detect all faults within PE1 or PE2 and therefore the resulting residual failures are also considered in the computation and are addressed by other measures (like periodic SW test, HW test or combination of both). This concept is shown in Figure 10 using a simplified FTA.

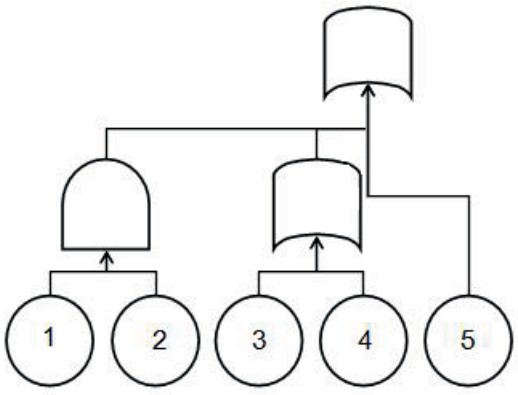

Figure 10 — Simplified FTA to show how to take credit of SW redundancy in ASIL decomposed MC

7.2.3 Coexistence of elements with different ASILs in MC components

This clause addresses safety related MC components implementing multiple coexisting requirements with different allocated ASILs (see reference[34]) and highlights some of the related requirements that can be relevant for MC components.

ISO 26262-9:2011, Clause 6 shall be considered and in particular according to ISO 26262-9:2011, 6.4.5, the rationale for freedom from interference is provided and supported by analyses of dependent

failures focused on cascading failures. See 7.2.4 of this document for further details on freedom from interference in MC components.

NOTE 1 Both systematic failures (in hardware or software) and random hardware failures have the potential to be initiators of dependent failures between elements with different ASILs. See 7.2.6 for further details. In such cases, independence of elements is ensured in accordance with ISO 26262-9:2011, Clause 7.

NOTE 2 The identification of initiators of dependent failures between elements with different ASIL can be supported by inductive or deductive analyses.

EXAMPLE In case of random hardware failures, similar parts or components with similar failure modes that appear several times in the FMEA can give additional information about the potential for dependent failures.

7.2.4 Freedom from interference (FFI) in MC components

If in a MC context multiple software elements with different ASIL ratings coexist, a freedom from interference analysis according to ISO 26262-9:2011, Clause 6 is carried out.

The exemplary faults listed in ISO 26262-6:2011, Annex D can be a starting point for the analysis.

NOTE 1 This clause focuses only on cascading faults between software elements implemented in PEs. Interferences can also be caused by HW dependent failures, in this case ISO 26262-9:2011, Clause 7 applies.

With respect to interference against “Memory” entries of ISO 26262-6:2011, D.2.3, the case of interference with private resources is considered. This type of interference can affect data or program regions belonging to one of the PEs.

EXAMPLE 1 Private data can be variables that belong to a safety-related software element in one of the PEs: A corruption of such variables from the other PEs leads to a malfunction of the software. In this case, a safety mechanism supervising the access and ensuring exclusive access helps to avoid interference. This example is related to SW interferences (i.e. the variable corruption is caused by a SW error). Interferences can also be caused by HW dependent failures, in this case ISO 26262-9:2011, Clause 7 applies.

EXAMPLE 2 Private program regions can be related to the corruption of a program in a non-volatile memory. In this case a mechanism restricting programming only from the higher ASIL elements helps to avoid interferences. This example can be applied to SW related interference (in a case where the program corruption is caused by a SW error; for example wrong permissions causing SW to overwrite the program memory). In this case ISO 26262-9:2011, Clause 7 applies.

This type of interference can also affect resources shared between different PEs.

EXAMPLE 3 A CAN peripheral is used by more than one core to exchange information with other ECUs. Interference can lead to an incorrect message transmission. In this case usage of robust end-to-end protection mechanisms (for example the ones listed in ISO 26262-5:2011, Table D.8) can help to detect interferences.

With respect to interference against “Time and execution” entries of ISO 26262-6:2011, D.2.2, the primary case to consider is interference that affects the execution latency of one core.

EXAMPLE 4 A CAN peripheral is used by more than one core to exchange information with other ECUs. If the PEs, processing tasks with a lower ASIL continuously request transmissions from the CAN peripheral then the higher ASIL tasks running in another core are not able to receive and/or transmit required information. A time monitoring mechanism (for example using the principles described for the safety mechanisms listed in ISO 26262-5:2011, Table D.10) can help to identify such conditions.

NOTE 2 Additional requirements related to timing are described in 7.2.7.

With respect to the interference against “Exchange of information” entries of ISO 26262-6:2011, D.2.4, interferences manifesting as failures in “Memory” or “Time and execution” can be caused by failures in exchange of information between different PEs.

EXAMPLE 5 A message from a non-safety related core is interpreted as safety related (masquerading fault).

NOTE 3 Usage of robust end-to-end protection mechanisms (for example the ones listed in ISO 26262-5:2011, Table D.8) can help to detect interference.

7.2.5 Software partitioning in MC components

When software partitioning, e.g. separation of functions or elements to avoid cascading failures, is used to implement freedom from interference between software components, ISO 26262-6:2011, 7.4.11 is applied.

Techniques such as hypervisors can help to achieve software partitioning (e.g. references[34] and[12]).

NOTE 1 Other techniques are also possible, such as microkernels (e.g. reference[12]).

It is worth considering the following points during safety analyses of MC involving hypervisors technologies:

— Virtualization technologies can support the argument to guarantee freedom from interference between software elements running in MC. A dependent failure analysis on software level is required and can be supported by consideration of the failure modes listed in ISO 26262-6:2011, Annex D.

NOTE 2 Positive effects of virtualization technologies with respect to freedom from interference can be compromised by systematic faults in hypervisor software. Similarly, virtualization technologies can be affected by hardware faults in the supporting hardware resources (like memory management unit) or in the related shared resources. Those faults are analysed according to the methods described in ISO 26262-9:2011, Clause 8 and dedicated guidance for integrated circuits is described in ISO 26262-10:2012, Annex A. Virtualization technologies can also be affected by HW dependent failures; in this case ISO 26262-9:2011, Clause 7 applies.

NOTE 3 If some of the hypervisor functions are delegated to tasks in the software partitions, then the analysis mentioned in Note 1 extends also to the partitions.

— Virtualization technologies are typically not able to provide sufficient prevention or detection of permanent or transient faults affecting the MC.

NOTE 4 Detection of specific hardware failure modes can be demonstrated by means of case by case detailed analyses based on the methods described in ISO 26262-9:2011, Clause 8. Dedicated guidance for integrated circuits is described in ISO 26262-10:2012, Annex A.

7.2.6 Dependent failures in MC component

Dependent failure analysis is carried out from both a hardware and software architectural point of view in accordance with ISO 26262-9:2011, Clause 7. For a dependent failure analysis of the hardware of a MC component, the methodologies defined in Clause 10 of this document can be applied.

7.2.7 Timing requirements in MC component

There are some clauses in ISO 26262-6 related to execution timing requirements, for example:

— 6.4.2 e) requires that the specification of the software safety requirements considers timing constraints;

— 7.4.17 a) requires that an upper estimation of required resources for the embedded software is made, including execution time;

— Table 13 Note c) shows that to ensure the fulfilment of requirements influenced by the hardware architectural design with sufficient tolerance, properties such as average and maximum processor performance, minimum or maximum execution times shall be determined;

— Annex D describes timing and execution failure modes (including incorrect allocation of execution time) as potential initiators of interferences between software elements.

MCs are potentially subject to timing faults (see reference[34]); therefore the previous listed clauses are carefully considered with dedicated analyses and adequate countermeasures identified.

EXAMPLE 1 Typical dedicated analyses for the identification of timing faults potentially violating the safety goal are based on the upper estimation of execution time (e.g. reference[13]).

EXAMPLE 2 Typical hardware-based countermeasures for detection of violation of timing requirements are watchdogs, timing supervision units and specific hardware circuits (e.g. reference[34]). Software-based countermeasures are also possible (e.g. reference[10]).

8 Programmable logic devices and ISO 26262

8.1 About programmable logic devices

8.1.1 General

As shown in Figure 11, PLDs can be seen as a combination of configurable I/O, non-fixed functions composed by logic blocks and user memory with a related configuration technology to configure them, signal routing capabilities connecting those logic blocks and fixed logic functions.

The non-fixed logic functions can include, but are not limited to, simple logic gates, multiplexers, inverters, flip-flops and memory to more complex functions such as digital signal processing functionality. Signal routing capabilities can range from simple point-to-point solutions, to complex bus interconnects with flexible routing possibilities and clocking options. PLDs can differ in their implementation of user memory. Some devices provide limited memory capabilities, while others provide local or global memory structures that can be used for a wide variety of applications. The more complex devices can also implement fixed functions such as CPUs, memory controllers, security modules, and others, thus freeing up design resources for user configurability. Clock, power and reset circuitries are fixed functions. It is up to the PLD design if single or multiple instances are implemented.

A common feature of PLDs is that users can configure them with the functionality adapted to the specific application needs. The design or configuration of the devices can be done with a variety of tools, ranging from very simple to entire development suites supporting complex features such as timing analysis and optimization of the design.

Once the user design is completed it can be programmed into the device. Different technologies are used to allow either one time programmability or the reprogramming of the device multiple times. These methods can be further distinguished by providing volatile or non-volatile capabilities. All of this is represented in the block diagram by the block labelled “configuration technology”.

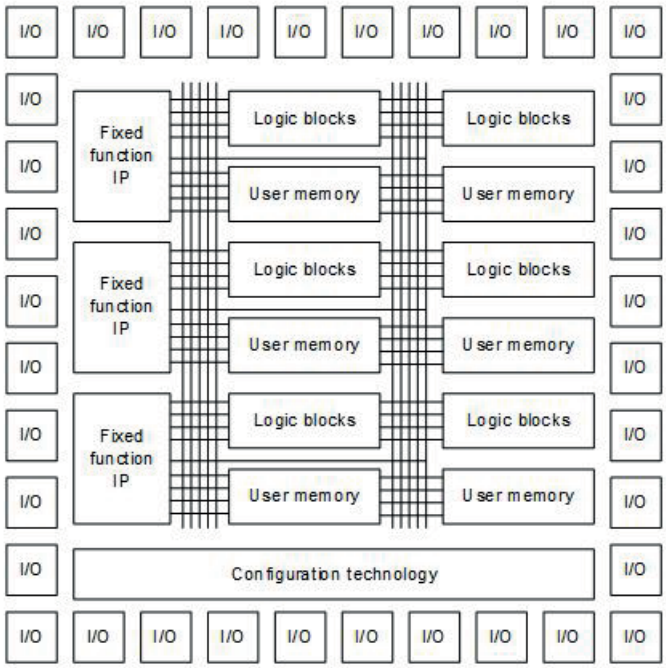

Figure 11 — A generic block diagram of a PLD

8.1.2 About PLD types

Table 16 provides a non-exhaustive list of commonly used PLD types.

Table 16 — Commonly used PLD types

Type	Description
Programmable **A**rray **L**ogic (PAL)	One-time programmable devices that allow implementing sum-of-products logic for each of its outputs.
Gate **A**rray **L**ogic (GAL)	Similar functionality as PALs with the feature of being programmable many times.
Complex **P**rogrammable **L**ogic **D**evice (CPLD)	Non-volatile devices with similar functionality as PALs with a much higher integration rate and additional complex feedback paths.
Field **P**rogrammable **G**ate **A**rray (FPGA)	Mostly volatile implementation of very sophisticated logic, routing and memory functions.

8.1.3 ISO 26262 Lifecycle mapping to PLD

8.1.3.1 General

Using the same structure of ISO 26262-10:2012, Figure 20, Figure 12 describes how the ISO 26262 lifecycle is mapping to PLDs.

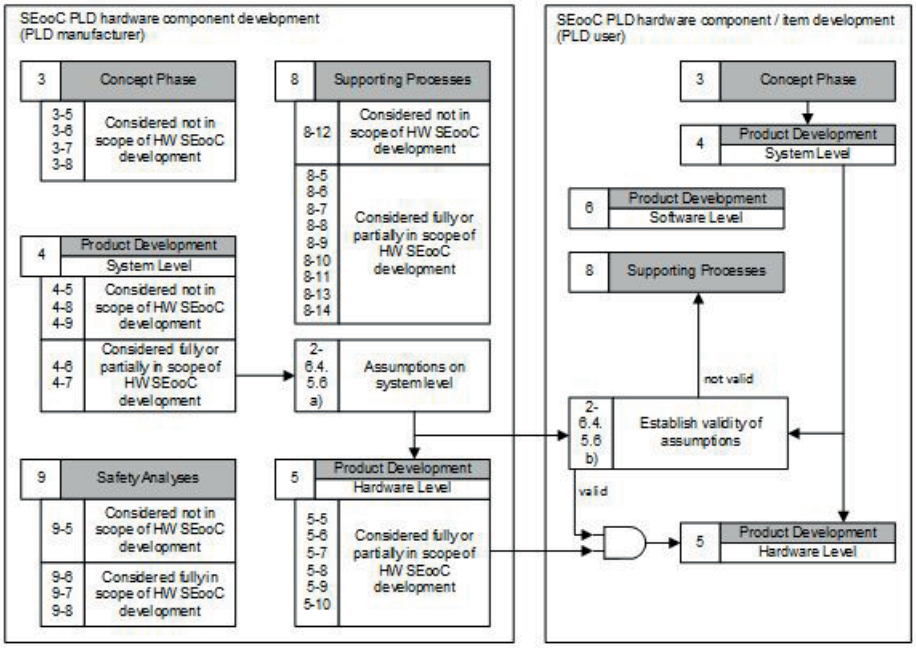

Figure 12 — ISO 26262 Lifecycle mapping to PLD

NOTE 1 In the context of this document, PLD manufacturer means an organization that develops the PLD and has the responsibility for the manufacturing of the PLD as semiconductor product. PLD user means an organization that develops a program for PLD or uses it in the application.

NOTE 2 Providers of IP blocks for PLD are considered in Clause 6.

NOTE 3 Although all the clauses of ISO 26262 are not shown in Figure 12, this does not imply that they are not applicable.

The following sections give examples with respect to some specific part of ISO 26262 for either PLD manufacturers or PLD users.

8.1.3.2 ISO 26262-2 (management of functional safety)

In general, ISO 26262-2 adapted to the appropriate level is applicable for the PLD manufacturer and the PLD user.

EXAMPLE 1 ISO 26262-2:2011, 6.4.2.1 requires that a project manager is appointed at the initiation of the item development. For a PLD manufacturer it means that a project manager is appointed at the initiation of the PLD development.

EXAMPLE 2 According to ISO 26262-2:2011, 6.4.3.5 the safety plan shall include the planning of the hazard analysis and risk assessment in accordance with ISO 26262-3:2011, Clause 7. Since the hazard analysis and risk assessment is done on item level only this requirement is not applicable for a safety plan on PLD level.

EXAMPLE 3 ISO 26262-2:2011, 6.4.8.1 requires a functional safety audit to be carried out for the item. Since it is not possible for the PLD manufacturer to carry out a safety audit on item level he carries it out on the PLD level instead.

EXAMPLE 4 ISO 26262-2:2011, 7.4.2.1 requires the organization to appoint persons with the responsibility and the corresponding authority, in accordance with ISO 26262-2:2011, 5.4.2.8, to maintain the functional safety of the item after its release for production. For a PLD manufacturer this means that a person is appointed to maintain the functional safety of the PLD after its release for production since he cannot be responsible for maintaining the functional safety of the whole item.

8.1.3.3 ISO 26262-3 (concept phase)

With respect to ISO 26262-3, the PLD manufacturer usually does not have any responsibility during the concept phase, unless the PLD manufacturer also assumes the role of item integrator. For the PLD user, this part is applicable if the PLD user also has responsibility at the item level.

8.1.3.4 ISO 26262-4 (product development at the system level)

For a SEooC development, ISO 26262-4:2011, Clause 6 and Clause 7 are partially or fully in scope. The same principle as discussed in ISO 26262-10:2012, 9.2.3 can be applied, where assumptions on the technical safety requirements and on the system-level design are made.

EXAMPLE Dedicated hardware safety measures can be implemented on the PLD by the PLD manufacturer to support the technical safety concept. Other measures can depend on the implemented user circuitry and can require specific measures (e.g. redundancy in logic, external watchdog) and are the responsibility of the user. The assumptions made by the PLD manufacturer on the system level measures is documented and verified by the PLD user.

If the PLD user is also the item integrator, ISO 26262-4 is fully in scope.

8.1.3.5 ISO 26262-5 (product development at the hardware level)

All the ISO 26262-5 clauses, including Clause 8 and Clause 9, are applicable to PLD manufacturers and PLD users according to their level of contribution to the overall safety concept.

EXAMPLE If the PLD does not include any HW safety mechanisms, the main role of PLD manufacturer is to provide base failure rate, failure modes, and failure modes distribution using, for example, the methods described in Clause 9 of this document. A reference or exemplary computation of hardware architectural metrics can be provided but the PLD user computes the metrics for the specific design the user implements in the PLD.

With respect to ISO 26262-5:2011, 5.8 and 5.9, the responsibility of PLD manufacturers is generally limited to providing the distribution of failure modes or the information/methods/tools needed to enable PLD users to compute/verify the metrics and to provide diagnostic coverage values for the safety mechanisms that are embedded in the PLD (see 8.4).

With respect to ISO 26262-5:2011, Clause 10, it is assumed in this document that it is not related only to integration tests but it is applicable as well to PLD manufacturers and PLD users testing activities according to their level of contribution to the overall safety concept. Regarding evaluation of the diagnostic coverage (ISO 26262-5:2011, Annex D), please refer to contents of 8.4.

8.1.3.6 ISO 26262-6 (product development at the software level)

Based on ISO 26262-4:2011, 7.4.5.2 and ISO 26262-5:2011, Clause 1, requirements of ISO 26262-5 and ISO 26262-6 can be combined in case of programmable logic like PLDs.

In case of a high-level synthesis flow, like developing in OpenCL, C-to-HDL flows, or a model based approach, interactions with the requirements of ISO 26262-6 are considered for the development of the high level language code. ISO 26262-5 is considered for follow on steps used for traditional PLD development.

In the case when the development flow for PLD users and PLD manufacturers is based on HDL languages, this is similar to the one used to develop microcontrollers, so ISO 26262-5 applies. ISO 26262-6 is not considered in this case.

NOTE Specific techniques and measures for user PLD circuit development are discussed in 8.5.3. For many methods there are similarities with respect to what is specified in ISO 26262-6, e.g. observation of coding guidelines.

The level of application of ISO 26262-6 also depends on the type of PLD technology. For example, in case of a PAL, the part is in general simple enough that ISO 26262-6 is not applied.

8.1.3.7 ISO 26262-7 (production and operation)

In general ISO 26262-7 adapted to the appropriate level is applicable for the PLD manufacturer. It is also applicable to the PLD user if he is involved in the production of a HW element of the item or of the item itself.

EXAMPLE 1 In ISO 26262-7:2011, 5.4.1.1 the requirement is to plan the production process by evaluating the item. In the context of the PLD manufacturer the planning is done by evaluating the PLD instead of the item.

EXAMPLE 2 ISO 26262-7:2011, 5.4.1.6 requires to identify reasonably foreseeable process failures and their effect on functional safety and to implement appropriate measure to address these issues. It is applicable to a PLD production without modification.

EXAMPLE 3 ISO 26262-7:2011, 6.4.1.5 requirements for decommissioning instructions are typically not applicable for PLDs

EXAMPLE 4 To fulfil ISO 26262-7:2011, 6.4.2.1 the PLD manufacturer implements a field monitoring process for the PLD.

8.1.3.8 ISO 26262-8 (supporting processes)

In general ISO 26262-8 adapted to the appropriate level is applicable for the PLD manufacturer and the PLD user.

With respect to ISO 26262-8:2011, Clause 13, the PLD can be either considered intermediate or complex part: this distinction is clarified in ISO/PAS 19451-2.

EXAMPLE A PAL can be considered to be an intermediate part, whereas a FPGA can be considered a complex part, according to ISO 26262-8:2011, Clause 13.

NOTE ISO 26262-8:2011, 5.4.1.1 b) applies to either basic or intermediate level hardware parts for which ISO 26262-8:2011, Clause 13 is applicable.

Regarding ISO 26262-8:2011, Clause 11, please refer to contents of 8.5.2.

8.1.3.9 ISO 26262-9 (Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses)

All ISO 26262-9 clauses are applicable to PLD manufacturers and PLD users according to their level of contribution to the overall safety concept.

Regarding ISO 26262-9:2011, Clause 7, please refer to contents of 8.3.2.

8.1.3.10 ISO 26262-10:2012 (Guideline on ISO 26262)

The contents of ISO 26262-10:2012, Annex A are applicable for PLD manufacturers and, in particular, for fixed function IPs. Further details about the applicability of ISO 26262-10:2012, Annex A are described in 8.3.1 and in 8.5.3.

8.2 Fault models and failure modes of PLD

In line with the lifecycle shown in 8.1.3, Table 17 and Table 18 summarize the fault models and the failure modes that can be of concern for PLD manufacturers and PLD users.

The listings do not claim exhaustiveness and can be adjusted based on additional known faults and failure modes. They can be used as a starting point to evaluate the diagnostic coverage of the provided safety mechanisms with the claimed DC. Any such claims are supported by a proper rationale.

NOTE 1 Table 17 and Table 18 address the same elements but on a different level of abstraction (fault models for PLD manufacturers and failure modes for PLD users). It is not intended that both will be considered at the same time (i.e. it is not intended that diagnostic coverages at both levels will be determined), but one of them will be considered depending on the respective abstraction level.

NOTE 2 Table 17 or Table 18 can also be used depending on the safety mechanism.

EXAMPLE For a redundant structure with a comparison, failure modes in Table 18 can be sufficient. For the evaluation of a self-test using test patterns, failure modes in Table 18 are not necessarily helpful in the evaluation of the diagnostic coverage and a detailed analysis per Table 17 can be performed.

Table 17 — Analysed faults in the derivation of diagnostic coverage for PLD manufacturers

Element (see Figure 11)	See tables	Analysed fault models for 60 %/90 %/99 % DC		
		Low (60 %)	**Medium (90 %)**	**High (99 %)**
Fixed Function IP	ISO 26262-5:2011, D.2 to D.14	As defined in ISO 26262-5:2011, Table D.1[a]		
PLD Digital I/O	ISO 26262-5:2011, D.7	As defined in ISO 26262-5:2011, Table D.1, element “Digital I/O”		
PLD Analogue I/O	ISO 26262-5:2011, D.7	As defined in ISO 26262-5:2011, Table D.1, element “Analogue I/O”		
Logic Block	Table 26	As defined in ISO 26262-5:2011, Table D.1, element “Other sub-elements not belonging to previous classes”		
Configuration Technology	Table 26	Stuck-at	Stuck-at at gate level Soft error model[b]	DC fault model Soft error model[b]
User Memory	ISO 26262-5:2011, D.6	As defined in ISO 26262-5:2011, Table D.1, element “Volatile memory”		
Signal Routing capability	ISO 26262-5:2011, D.14	As defined in ISO 26262-5:2011, Table D.1, element “On-chip communication including bus-arbitration”		

[a] As described in 8.1, the Fixed Function IPs are a combination of elements similar to the ones that can be found in microcontrollers. They are typically implemented in a separated area with respect to the non-fixed functions and therefore they can be considered in all aspects similar to the elements discussed in ISO 26262-5:2011, Table D.1 and ISO 26262-10:2012, Table A.1

[b] The relevance of this fault model depends on the type of PLD technology, see 8.1.2.

Table 18 — Analysed failure mode in the derivation of diagnostic coverage for PLD users

Element (see Figure 11)	See tables	Analysed failure modes for 60 %/90 %/99 % DC		
		Low (60 %)	**Medium (90 %)**	**High (99 %)**
Fixed Function IP	ISO 26262-5:2011 D.2 to D.14	As defined in ISO 26262-5:2011, Table D.1[a]		
PLD Digital I/O	ISO 26262-5:2011, D.7	As defined in ISO 26262-5:2011, Table D.1, element “Digital I/O”		
PLD Analogue I/O	ISO 26262-5:2011 D.7	As defined in ISO 26262-5:2011, Table D.1, element “Analogue I/O”		

[a] As described in 8.1, the fixed function IPs are a combination of elements similar to the ones that can be found in microcontrollers. They are typically implemented in a separated area with respect to the non-fixed functions and therefore they can be considered in all aspects similar to the elements discussed in ISO 26262-5:2011, Table D.1 and ISO 26262-10:2012, Table A.1

[b] The relevance of this failure mode depends on the type of PLD technology and type of Logic Block, see 8.1.2.

[c] The relevance of this failure mode depends on the type of PLD technology, see 8.1.2.

[d] The I/O configuration logic can be inside the fixed function IP or in the I/O itself.

Table 18 *(continued)*

Element (see Figure 11)	See tables	Analysed failure modes for 60 %/90 %/99 % DC		
		Low (60 %)	**Medium (90 %)**	**High (99 %)**
Logic Block	Table 26	Permanent corruption of the function implemented by the logic block	Permanent corruption of the function implemented by the logic block. Transient corruption of the function implemented by the logic block.	Permanent corruption of the function implemented by the logic block. Transient corruption of the function implemented by the logic block.
Configuration Technology	Table 26	Unintentional permanent change of the configuration settings	Unintentional permanent change of the configuration of the logic block. Unintentional transient change of the configuration of one logic block. [c]	Unintentional permanent change of the configuration of the logic block. Unintentional transient change of the configuration of one logic block. [c]
User Memory	ISO 26262-5:2011, D.6	As defined in ISO 26262-5:2011, Table D.1, element "Volatile memory"		
Signal Routing capability	ISO 26262-5:2011, D.14	Permanent corruption of the function implemented by a group of logic blocks	Permanent corruption of the function implemented by a group of logic blocks, including time out of the function. Transient corruption of the function implemented by a group of logic blocks.	Permanent corruption of the function implemented by a group of logic blocks, including time delay of the function. Transient corruption of the function implemented by a group of logic blocks.

[a] As described in 8.1, the fixed function IPs are a combination of elements similar to the ones that can be found in microcontrollers. They are typically implemented in a separated area with respect to the non-fixed functions and therefore they can be considered in all aspects similar to the elements discussed in ISO 26262-5:2011, Table D.1 and ISO 26262-10:2012, Table A.1

[b] The relevance of this failure mode depends on the type of PLD technology and type of Logic Block, see 8.1.2.

[c] The relevance of this failure mode depends on the type of PLD technology, see 8.1.2.

[d] The I/O configuration logic can be inside the fixed function IP or in the I/O itself.

8.3 Notes about safety analyses for PLDs

8.3.1 Quantitative analysis for a PLD

A similar approach as discussed in ISO 26262-10:2012, Annex A can be also used for PLDs. A quantitative analysis of the PLD including the user design can be performed on different abstraction levels depending on the information available to the PLD user. Information about the PLD usage and user design is refined during the development phase of the design and the analysis is repeated based on the latest information. The quantitative analysis of the PLD design can be augmented by a dependent failure analysis as described in 8.3.2.

The following two sections describe examples of PLD die failure rate calculations and examples of the distribution of the failure rate to the identified failure modes.

The hardware architectural metrics can be determined similar to the example given in ISO 26262-10:2012, A.3.5. The level of detail required for the analysis depends on the targeted ASIL and the application.

8.3.1.1 Example of PLD die failure rate calculation per IEC/TR 62380

The failure rates can be estimated as described in ISO 26262-5:2011, 8.4.3.

NOTE If failure rates provided by the PLD manufacturer are used, any de-rating factor applied to the provided data are made available.

This example follows the example given in ISO 26262-10:2012, A.3.4.2. It makes similar assumptions and not all notes are repeated in this section. A PLD with the characteristics outlined in Table 19 is used for the example.

Table 19 — PLD resource overview

Element	Resources	Assumed IEC/TR 62380 category
Logic blocks	1 000	CPLD (EPLD, MAX, FLEX, FPGA, etc.)
User memory	b	Low-consumption SRAM
Fixed function IP	20 k gates	Digital circuits, microcontroller, DSP
Configuration technology	10 kb	Low-consumption SRAM

Similar to the example given in ISO 26262-10:2012, Table A.2, the complete PLD failure rate can be computed as shown in Table 20.

Table 20 — Example of the computation of the failure rates for the PLD

Element	λ_1	N	α	λ_2	Base FIT	De-rating for temp	Effective FIT
Logic blocks	$2{,}0 \times 10^{-5}$	100 000 (100 transistors per macrocell)	10	34	34,0604	0,17	5,7903
User memory	$1{,}7 \times 10^{-7}$	98 304 (6 transistors/bit for a low-consumption SRAM)	10	8,8	8,8005	0,17	1,4961
Fixed function IP	$3{,}4 \times 10^{-6}$	80 000 (4 transistors / gate)	10	1,7	1,7082	0,17	0,2904
Configuration technology (based on SRAM)	$1{,}7 \times 10^{-7}$	61 440 (6 transistors/bit for a low-consumption SRAM)	10	8,8	8,8003	0,17	1,4961
Sum					53,3694		9,0729

NOTE 1 It is assumed that the number of transistors per macrocell (100, as derived from IEC/TR 62380) does not include the transistors related to the configuration technology. For this reason the configuration technology is considered as a separate entry of the computation. An alternative approach could be to adapt the number of transistors and include the configuration technology in the logic blocks, user memory entries and other relevant elements.

NOTE 2 1 FIT corresponds to 1 failure per 10^9 h of device operation

NOTE 3 This table can be used also to derive a unitary FIT by dividing the resulting effective FIT with the number of elements.

EXAMPLE The FIT/logic block can be computed as 5,7903/1 000 = 0,0057

NOTE 4 As shown in ISO 26262-10:2012, A.3.4.2.2 other alternatives are possible for the temperature de-rating factor. Those alternatives are applicable as well for PLDs.

The failure rates in Table 20 can be used to calculate the failure rates for this specific user design. The assumptions made for the user design are given in Table 21.

Table 21 — Example of user design resource usage and failure rate calculation

Element	Resource usage	Effective FIT
Logic blocks	23 %	1,3318
User memory	10 %	0,1496
Fixed function IP	100 %	0,2904
Configuration technology (based on SRAM)	15 %	0,2244
Sum		1,9962
NOTE 1 The unused resources are considered as not safety related. Depending on the PLD structure, a dependent failure analysis can analyse the influence of the unused logic on the user design. NOTE 2 An alternative approach is to consider the unused logic as safety related and to estimate the respective fraction of faults that will lead to a safe failure (F_{safe} according ISO 26262-10:2012, Figure 9). This estimation can be done by means of a quantitative analysis supported by information provided by the PLD manufacturer.		

The data can be further refined if more detail about the user design is available. For example a logic block has different configuration options and the user design may only use a certain configuration. This allows to further de-rate the calculated failure rate.

NOTE 1 A dependent failure analysis can be used to analyse the influence of the different configuration options on the user design.

NOTE 2 The derivation of the de-rating factor can be facilitated by appropriate design tools.

8.3.1.2 Example of transient failure rate calculation for PLD

The computation of the transient failure rate for PLD can follow ISO 26262-10:2012, A.3.4.1, i.e. considering data provided by the PLD manufacturer derived from JEDEC standards such as JESD 89A or, if this data are not available, soft error rate derived from public sources such as International Technology Roadmap for Semiconductors (ITRS).

NOTE In case the transient failure rate provided by the PLD manufacturer includes a de-rating factor (for example based on average PLD utilisation factor or based on operational profile), this factor is explained to the PLD user.

Table 21 can be used to calculate the failure rates for this specific user design in the same way for transient faults, as shown in the previous paragraph.

8.3.1.3 Example of distribution of PLD failure rate to failure modes

Once the PLD failure rate has been estimated, it is distributed to the identified failure modes, i.e. the failure modes distribution is computed.

For PLD manufacturers, the failure modes distribution can be computed as described in ISO 26262-10:2012, Annex A.

The following are examples of approaches for identification of failure modes and respective determination of the failure modes distribution for PLD users:

a) Identification of the failure modes at the functional block level of the user PLD design; assumption of an equal distribution of the PLD failure rate to the identified failure modes;

b) Identification of the failure modes at the functional block level of the user PLD design; estimation of the distribution of the PLD failure rate to the identified failure modes based on expert judgment taking resource estimation (e.g. fixed function IP, number of logic blocks, user memory, etc.) into account, supported by documented evidences;

c) Identification of the failure modes by means of a partitioning of the implemented user PLD design in elementary sub-parts; estimation of the distribution of the PLD failure rate to the identified failure modes based on the implemented user PLD design facilitated by information provided by the PLD manufacturer taking detailed resource utilisation into account. This could be supported by appropriate design tools.

NOTE 1 Elementary sub-part is defined as in the example in ISO 26262-10:2012, 4.2. In the context of PLD manufacturer, the elementary sub-part can be intended as a set of flip-flop and gates (e.g. logic cone). At the same way, in the context of PLD users, the elementary sub-part can be intended as the cone constructed of flip-flop in a logic block and the combinatorial logic represented by logic blocks. The level of detail, i.e. the number of elementary sub-parts considered depends on the targeted ASIL, the type of safety mechanism used and the application.

NOTE 2 The level of accuracy of the resulting quantitative data varies based on the approach used.

EXAMPLE 1 If information on the implemented user PLD design is available, then approach c) can provide the highest level of accuracy. If this information is not available and no argument can be given why one of the failure modes is more likely than the other, the approach a) can be used.

NOTE 3 The required level of accuracy of the failure mode distribution depends also on the targeted ASIL, the type of safety mechanism used and the application.

EXAMPLE 2 In case of a user PLD design in lock-step, approach a) can be sufficient because a non-uniform distributed value for the failure mode distribution will not affect the claimed diagnostic coverage. Instead, for a user PLD design relying on a SW test library to periodically test the PLD hardware, if arguments exist that one of the failure modes is more likely than the other approaches b) or c) are used depending on the required level of accuracy.

NOTE 4 A detailed failure mode definition like the one provided by approach c) can help to provide rationale for diagnostic coverage.

NOTE 5 For transient faults, the resource utilisation can consider number of flip flops included in the logic blocks and the number of user memory bits of the user PLD design and number of configuration bits utilized by the user PLD design

Table 22 shows an example of the three approaches described above. It considers a SPI module implemented in a PLD.

Table 22 — Example of approaches for PLD failure modes distribution computation at PLD user level

Failure mode	Sub-parts involved	a)	b) See note 1	c) See note 2
Wrong or no clock	Clock generation	25 %	10/110 = 9,09 %	10/90 = 11,11 %
Wrong or no data reception	Peripheral bus interface Input shift register Data received register I/O pads	25 %	40/110 = 36,36 %	30/90 = 33,33 %
Wrong or no data send	Peripheral bus interface Output shift register Data send register I/O pads	25 %	40/110 = 36,36 %	30/90 = 33,33 %
Wrong configuration of SPI	Configuration registers Peripheral bus interface	25 %	20/110 = 18,18 %	20/90 = 22,22 %

NOTE 1 For this example, it is estimated that each sub-part consumes 10 logic blocks and therefore it is estimated that each failure mode has a failure mode distribution proportional to the sum of logic blocks consumed by each sub-part involved in the failure mode

NOTE 2 The difference between b) and c) is that the resource usage for the specific failure mode is not estimated anymore but the actual number of resources which contribute to the failure mode is computed. This is done not necessarily only on the sub-part level but also down to the elementary sub-parts level, if the logic blocks contributing to the failure mode span different sub-parts. In the example, it is measured that: Input shift register, output shift register, data received register and data send register are contributing 100 % to the respective failure mode and 0 % to the others; peripheral bus interface is measured to contribute 50 % to each data related failure mode and 100 % to configuration failure mode; I/O pads are measured to contribute 50 % to each data related failure mode.

8.3.1.4 Verification of completeness and correctness of safety mechanism implementation with respect to hardware

As described in ISO 26262-10:2012, A.3.8.2, fault injection simulation during development phase is a valid method to verify completeness and correctness of safety mechanism implementation with respect to hardware safety requirements as also to assist verification of safe faults and computation of their amount and failure mode coverage, as described in ISO 26262-10:2012, A.3.3 and A.3.3.2. This applies for PLD manufacturers as well.

With respect to PLD users, in case fault injection is necessary and no detailed information is available about how the user PLD design is mapped to PLD logic blocks, fault injection can be performed on the logic design before mapping.

EXAMPLE If fault injection is necessary to provide rationale of the diagnostic coverage claimed by a SW test library periodically testing the user PLD design, then fault injection can be executed at a different level. For example, starting from the RTL design describing the user PLD design and then synthesizing it to obtain a reference netlist on which fault injection is performed. If the reference netlist does not correspond to the PLD design, then an argumentation is provided to explain why the injected faults are meaningful with respect to the assumed implementation of PLD design.

8.3.2 Dependent failure analysis for a PLD

As for any integrated circuit, dependent failures are important to be considered especially if HW safety mechanisms or requirements for redundancy are implemented in the same component.

The flow for Dependent Failure Analysis (DFA) considered in this clause is the same than the one described in Clause 10. Table 23 describes specificities – if any – to be considered in addition with respect to the steps defined in Clause 10, for both PLD manufacturer and PLD users.

Table 23 — Specificities of DFA for PLD manufacturers and PLD users with respect to Clause 10

Step (see Figure 24)	**PLD manufacturer**	**PLD user**
B1 – Identify HW and SW elements	As defined in Clause 10	As defined in Clause 10
B2 – Identify dependent failures initiators	Analysis considers also the interactions between configurable and fixed logic, including interactions related to reset or the configuration technology[a]	Analysis considers also the impact of failures affecting the configuration technology and therefore potentially affecting multiple logic blocks at the same time
B6 – Identify necessary safety measures to control or mitigate dependent failures initiators	Analysis considers also the possibilities for providing separation between configurable and fixed logic	Analysis considers also the possibilities for providing separation between logic blocks
B10 – Evaluate the effectiveness to control or to avoid the dependent failure	As defined in Clause 10	As defined in Clause 10
[a] For example, a fault in the fixed logic causing the configurable logic to lose the configuration		

The list for dependent failure initiators (DFI) considered in this document is the same than the one described in Clause 10. The following tables describe specificities – if any – to be considered in addition with respect to DFI defined in Clause 10, for both PLD manufacturer and PLD users, and the related countermeasures.

Table 24 — Specificities of DFI for PLD manufacturer and PLD user with respect to Clause 10

Dependent Failure Initiators (DFI)	**PLD manufacturer DFI**	**PLD user DFI**
Failure of shared resources [a]	As defined in Clause 10	Potential dependency of the available clock networks Failures of configuration technology (e.g. shared short or long distance common interconnects) Failures of shared programmable I/Os Wrong PLD configuration due to failures of external configuration memory or related interconnection
Single physical root cause	As defined in Clause 10	Faults (e.g. in reset logic) causing the complete or partial loss of the PLD configuration
Development faults	Insufficient distance or isolation between fixed and configurable logic	Wrong usage of tools provided by PLD manufacturer [b] See also Clause 10
Manufacturing faults	As defined in Clause 10	Wrong usage of tools for configuration programming [b]
Installation faults	As defined in Clause 10	As defined in Clause 10
Repair faults	As defined in Clause 10	Wrong usage of online reconfiguration functions
[a] In the context of PLD, "common" will be interpreted not only as shared resources within either configurable or fixed logic but also as shared resources between configurable and fixed logic. [b] For example, user wrongly applies isolation/separation constraints		

Table 25 — Countermeasures related to DFI for PLD manufacturer and PLD user

Dependent Failure Initiators (DFI)	PLD manufacturer counter-measures	PLD user countermeasures
Failure of shared resources [a]	As defined in Clause 10	Analysis of dependency of clock networks and dedicated clock monitors Analysis of failures of configuration technology and consequent adoption of separation/isolation techniques Analysis of failures of shared programmable I/Os and consequent adaptation of I/Os safety protocols CRC check of PLD configuration during runtime
Single physical root cause	As defined in Clause 10	Analysis of dependency of the reset networks and dedicated watchdogs
Development faults	Proper isolation or separation between fixed and configurable logic	As defined in Clause 10
Manufacturing faults	As defined in Clause 10	Proper instructions in PLD tool manual to prevent DFI
Installation faults	As defined in Clause 10	As defined in Clause 10
Repair faults	As defined in Clause 10	Restricted use of online reconfiguration functions

[a] In the context of PLD, "common" will be interpreted not only as shared resources within either configurable or fixed logic but also as shared resources between configurable and fixed logic.

8.4 Examples of safety mechanisms for PLD

Table 26 lists examples of safety mechanisms defined in ISO 26262-5:2011, Annex D that can be used to address PLD failure modes described in Table 17 and Table 18. Table 26 also contains additional safety mechanisms not included in ISO 26262-5:2011, Annex D that can be applied to PLDs. This table is not exhaustive and other techniques can be used, provided evidence is available to support the claimed diagnostic coverage.

Table 26 — Mapping of PLD safety mechanisms with ISO 26262-5:2011, Annex D

Element	Examples of safety mechanisms
Fixed function IP	ISO 26262-5:2011, Tables D.4 to D.13
Clock	ISO 26262-5:2011, Table D.10 On-chip clock status indication[a]
Power supply	ISO 26262-5:2011, Table D.9 Separate voltage planes [b]
Digital I/O	ISO 26262-5:2011, Table D.7
Analogue I/O	ISO 26262-5:2011, Table D.7

[a] Many PLDs offer clock generation and management resources and also provide monitoring of clock functionality and associated status pins/register to indicate when a specific clock is functioning properly (e.g. whether or not a clock output is in proper phase with a master clock input).

[b] Voltage plane means electrically isolated voltage supply plane regions with each plane region being connectable to an external supply voltage.

[c] Refers to the capability of many programmable devices to check the contents of its configuration registers and compare those to the intended (design specific) contents. If a mismatch is detected, this feature can change the status of an output pin or generate an interrupt so that the system can respond appropriately.

Table 26 *(continued)*

Element	Examples of safety mechanisms
Logic block	ISO 26262-5:2011, Tables D.4 and D.13 Mix of spatial and temporal redundancy by means of reconfiguration
Off-chip communication	ISO 26262-5:2011, Tables D.7 and D.8
Configuration technology	ISO 26262-5:2011, Table D.5 (non-volatile) and/or Table D.6 (Volatile) Read-back on download by downloading device [c]
User memory	ISO 26262-5:2011, Table D.5 (non-volatile) and/or Table D.6 (Volatile)
Signal routing capability	ISO 26262-5:2011, Table D.14

[a] Many PLDs offer clock generation and management resources and also provide monitoring of clock functionality and associated status pins/register to indicate when a specific clock is functioning properly (e.g. whether or not a clock output is in proper phase with a master clock input).

[b] Voltage plane means electrically isolated voltage supply plane regions with each plane region being connectable to an external supply voltage.

[c] Refers to the capability of many programmable devices to check the contents of its configuration registers and compare those to the intended (design specific) contents. If a mismatch is detected, this feature can change the status of an output pin or generate an interrupt so that the system can respond appropriately.

8.5 Avoidance of systematic faults for PLD

8.5.1 Avoiding systematic faults in the implementation of PLD

Since there are no significant differences in the specification, design and verification flow used by PLD manufacturers with respect to the flow used by microcontroller manufacturers, the same recommendations given in ISO 26262-10:2012, A.3.7 (and related Table A.8) can be applied.

8.5.2 About PLD supporting tools

PLD related tools can be distinguished in two categories:

— tools used prior to the production (i.e. used by PLD manufacturers);

— tools used by PLD users

The confidence in use of tools belonging to both categories are analysed according to the requirements of ISO 26262-8:2011, Clause 11.

EXAMPLE 1 According ISO 26262-8:2011, Clause 11, a tool used for place and route by the PLD manufacturer can be considered TI2, since its malfunction can introduce an errors in a safety-related element being developed; If it can be shown that design rule check (DRC) and layout versus schematic (LVS) with appropriate rule sets, as foreseen in state-of-the-art IC design flows, can detect possible errors introduced by the tool with a high degree of confidence, then a TD1 can be claimed. In this case it can be considered as TCL1 based on ISO 26262-8:2011, Table 3.

EXAMPLE 2 According ISO 26262-8:2011, Clause 11, a tool used for place and route by the PLD users can be considered TI2, since its malfunction can introduce an error in a safety-related element being developed. If the error can be detected with a medium degree of confidence by the consequent HW and integration tests, due to the complexity of the circuitry, then it can be considered TD2. Therefore it can be considered as TCL2 based on the ISO 26262-8:2011, Table 3. If the ASIL of the respective item is for example ASIL B, the tool provider can qualify the SW tool by using an appropriate combination of “increased confidence from use” and “evaluation of the tool development process”.

8.5.3 Avoiding systematic faults for PLD users

For PLD manufacturers, as for a microcontroller, a PLD is developed based on a standardized development process for which the example in ISO 26262-10:2012, A.3.7 applies.

The two following approaches are examples of how to provide evidence that sufficient measures for avoidance of systematic failures are taken care of by the PLD user during the development,, by using appropriate processes:

— using a checklist such as the one reported in Table 27; and

— giving the rationale by field data of similar products which are developed based on the same process as the target device (for example using ISO 26262-8:2011, Clause 14).

Table 27 — Examples of measures to avoid systematic failures for PLD users

ISO 26262-5 requirement	Design phase	Technique/Measure	Aim
7.4.1.6 Modular design properties	**Design entry**	Structured description and modularization	The description of the PLDs functionality is structured in such a fashion that it is easily readable, i.e. circuit function can be intuitively understood on basis of description without simulation efforts
7.4.1.6 Modular design properties		Design description in HDL	Functional description at high level in hardware description language, for example such like VHDL or Verilog.
Robust design principles		Observation of coding guidelines	Strict observation of the coding style results in a syntactic and semantic correct circuit code
Robust design principles		Restricted use of asynchronous constructs	Avoidance of typical timing anomalies during synthesis, avoidance of ambiguity during simulation and synthesis caused by insufficient modelling, design for testability. This does not exclude that for certain types of PLD implementations, asynchronous logic could be useful; in this case, the aim is to suggest additional care to handle and verify those circuits.
Robust design principles		Synchronisation of primary inputs and control of metastability	Avoidance of ambiguous circuit behaviour as a result of set-up and hold timing violation
7.4.4 Verification of HW design		HDL simulation	Functional verification of circuit described in VHDL or Verilog by means of simulation
7.4.4 Verification of HW design		Functional test on module level (using for example HDL test benches)	Functional verification “Bottom-up”
7.4.4 Verification of HW design		Functional test on top level	Verification of the PLD (entire function)
7.4.4 Verification of HW design		Functional and structural coverage-driven verification (with coverage of verification goals in percentage)	Quantitative assessment of the applied verification scenarios during the functional test. The target level of coverage is defined and shown
7.4.4 Verification of HW design		Application of code checker	Automatic verification of coding rules (“coding style”) by code checker tool.
7.4.4 Verification of HW design		Documentation of simulation results	Documentation of each data needed for a successful simulation in order to verify the specified circuit function.
7.4.4 Verification of HW design		Integration and verification of soft IPs	See Clause 6

Table 27 *(continued)*

ISO 26262-5 requirement	Design phase	Technique/Measure	Aim
7.4.4 Verification of HW design	**Synthesis, mapping, floor planning, placement, routing**	Check of PLD vendor requirements and constraints	Requirements and constraints defined by PLD vendor are considered during PLD design
7.4.4 Verification of HW design		Analysis of PLD supporting tool outputs	Outputs of PLD supporting tools are analysed. Arguments are provided to waive warnings and Errors.
7.4.1.6 Modular design properties		Documentation of constraints, results and tools	Documentation of each defined constraint that is necessary for an optimal synthesis, mapping, placement and routing of the PLD design
7.4.1.6 Modular design properties		Script based procedures	Reproducibility of results and automation of the synthesis, mapping, placement and routing
7.4.4 Verification of HW design		Simulation and timing verification of the final netlist	Independent verification of the netlist after synthesis, mapping, placement and routing – including timing verification
7.4.4 Verification of HW design		Comparison of the final netlist with the reference model (formal equivalence check)	Functional equivalence check of the final netlist with RTL.
Robust design principles		Adequate time margin for process technologies in use for less than three years	Assurance of the robustness of the implemented circuit functionality even under strong process and parameter fluctuation. A time margin in the timing analysis is considered either in the libraries or by PLD user.
7.4.4 Verification of HW design		Design rule check (DRC)	Execution of design rule checks on the floor planned I/O logic
9.4.2.4 Dedicated measures 10 Hardware integration and testing	**PLD integration and testing**	PLD verification	Verification of the PLD prototype, including verification of PLD correct configuration (e.g. using checksums).
7.4.5 Production, operation, service and decommissioning 9.4.2.4 Dedicated measures 10 Hardware integration and testing		PLD integration	Verification and integration of the PLD in the system

8.6 Safety documentation for a PLD

ISO 26262-10:2012, A.3.10 gives recommendations in terms of the safety documentation for a SEooC microcontroller and in terms of the contents of the documentation which may be consolidated in a so called “Safety Manual” or “Safety Application Note”. Those recommendations can be used also by PLD manufacturers and PLD users, with the following remarks:

— The DIA between PLD manufacturer and PLD user specifies which documents are made available and what level of detail is provided to the PLD user;

— The main focus of the safety documentation provided by PLD manufacturer is:

— the description of the results of the analyses of the development processes of the PLD manufacturer with respect to the applicable requirements of ISO 26262;

— the description of the results of the analyses of the PLD supporting tools with respect to the applicable requirements of ISO 26262;

— the provision of information (for example the PLD failure rate, the PLD failure modes with the related failure modes distribution, the claimed diagnostic coverage for safety mechanisms that are already implemented in the PLD etc.) to be used by PLD users during their safety analyses;

— proposals or examples of safety mechanisms, for example with respect to dependent failures etc.;

— the list of assumptions of use to guide PLD users in the correct utilisation of the safety-related information provided with the PLD.

— The work products of the safety lifecycle are provided by the PLD user. The completeness of the work products depends on whether the PLD user also assumes the role of the item integrator.

8.7 Example of safety analysis for PLD

8.7.1 Architecture of the example

Figure 13 is an example system used to demonstrate the concepts outlined in this paper. The system is intended for a safety critical application where two microcontrollers are used for redundancy and the final control output is implemented using a PLD. The two microcontrollers send their values to the PLD via SPI (Serial Peripheral Interface) and the PLD communicates its output via a CAN (Controller Area Network) bus. For this example, it is assumed that a calculated output too high (i.e. greater than the value that would have been determined by a non-faulted system plus a threshold) is a potential hazard but an output too low is acceptable from a functional safety point-of-view. It is also assumed that the components receiving the CAN message can detect the loss of CAN messages and take appropriate remedial action such as defaulting the receive signal to its minimum value and that the receiving module can tolerate corrupted CAN messages (i.e. values higher than intended) for x number of messages.

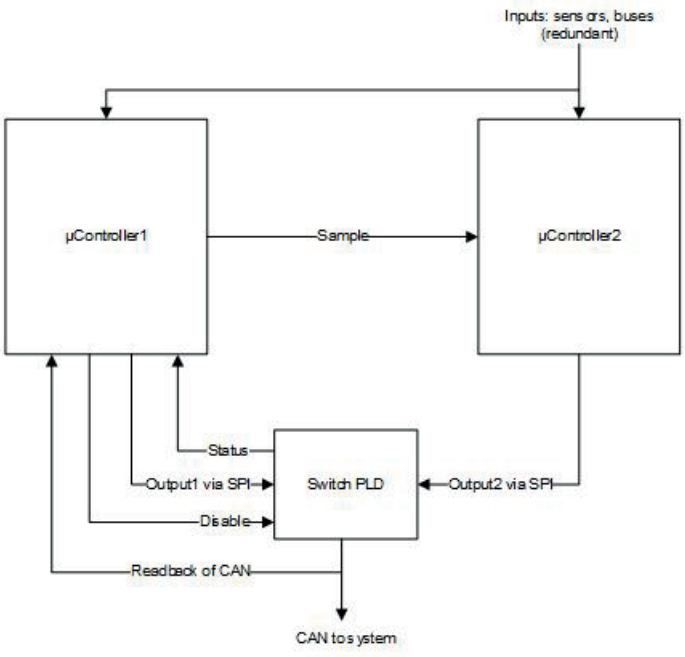

Figure 13 — Example of PLD usage – output switch

NOTE The HW component "Controller" is implemented using two microcontrollers and one PLD.

The derived safety requirement for the HW component “Controller” could be:

— SafReq_HW_Comp_Controller_001: The output of a wrong value which is larger than the correct value plus a threshold for x number of messages in-a-row shall be avoided.

— SafReq_HW_Comp_Controller_002: Undetected lack of CAN outputs for longer than y ms shall be avoided.

The HW component “Controller” is implemented using two microcontrollers (µController1 and µController2) and one PLD. Both µController1 and µController2 have the same input/output history and send their calculated outputs to the PLD. Both outputs agree within the threshold when no fault has occurred. The PLD is responsible for taking the minimum of the two signals and communicating this output to the rest of the system via CAN. SafReq_HW_Comp_Controller_002 can be fulfilled by entities outside of the controller (e.g. timeout supervision).

The derived safety requirement for the PLD could be:

— SafReq_PLD_001: Output of a value larger than the minimum of the two input values from µController1 and µController2 shall be avoided (derived from SafReq_HW_Comp_Controller_001).

— SafReq_PLD_002: Undetected corruption of the CAN output value from PLD which leads to an output too high shall be avoided (derived from SafReq_HW_Comp_Controller_001).

The following section addresses, as an example, two different approaches for the PLD’s safety and dependent failure analysis. The safety analysis and the dependent failure analysis concerning µController1 and µController2 are out of scope of this document.

Failures of the PLD can be addressed by two approaches:

— Utilizing safety measures which are external to the PLD

— Utilizing safety measures which are internal to the PLD. The PLD includes diagnostic measures to detect faults of the PLD. Faults are communicated via the status signal to µController1, which can disable the PLD based on the severity of the fault.

8.7.2 PLD external measures

The following safety mechanisms are implemented by elements other than the PLD:

— SafMech_PLD_001: CAN Read back and comparison. The CAN output of the PLD is read back by µController1. µController1 checks if the PLD has output a value equal or less than its output. If this check fails the µController1 disables the PLD via the Disable signal.

— SafMech_Network_001: The receivers implement a time-out monitoring.

As a first step of the safety analysis the relevant failure modes can be identified. Since none of the safety mechanisms are implemented within the PLD it is sufficient to describe the observable failure modes on its output level:

— FM_PLD_OP_01: No output;

— FM_PLD_OP_02: Output of old message;

— FM_PLD_OP_03: Corrupt output;

— FM_PLD_OP_04: Do not output minimum value;

— FM_PLD_OP_05: Always output µController1 value;

— FM_PLD_OP_06: Always output µController2 value;

— FM_PLD_OP_07: Active “Disable” discrete signal does not prevent CAN transmission.

As described in 8.3.1.3, to derive a probability distribution over the above mentioned failure modes typically detailed knowledge of the PLD internal structure is necessary. If this information is not available and no argument can be given why one of the failure modes is more likely than the other, the approach described in 8.3.1.3 a) can be adopted. In this case the safety analysis could be similar to the one in Table 28.

Table 28 — Example of a PLD safety analysis in case of PLD external measures

Failure mode	Permanent distribution	Transient distribution	PVSG?	MPF?	Safety mechanisms
FM_PLD_OP_01: No output	14,2 %	14,2 %	1	0	SafMech_Network_001
FM_PLD_OP_02: Output of old message	14,2 %	14,2 %	1	0	SafMech_PLD_001
FM_PLD_OP_03: Corrupt output	14,2 %	14,2 %	1	0	SafMech_PLD_001
FM_PLD_OP_04: Do not output minimum value	14,2 %	14,2 %	0	1	SafMech_PLD_001
FM_PLD_OP_05: Always output µController1 value	14,2 %	14,2 %	0	1	
FM_PLD_OP_06: Always output µController2 value	14,2 %	14,2 %	0	1	SafMech_PLD_001
FM_PLD_OP_07: Active "Disable" discrete signal does not prevent CAN transmission	14,2 %	14,2 %	0	1	
NOTE PVSG = potential to directly violate the safety goal; MPF = multiple point failure					

As far as the dependent failure analysis (out of scope of this document) is concerned the correlation of following elements could be of interest:

— PLD and µController1;

— PLD and µController2;

— µController1 and µController2;

8.7.3 PLD internal measures

The rest of the example considers utilizing safety measures which are internal to the PLD. The internal architecture of the PLD is presented in Figure 14. The data sent from the µController should be buffered before it can be transferred via the CAN bus. The buffers are implemented as user memory, whereas the state machine controlling the buffer operation, the multiplexer are implemented by logic blocks and the CAN module is a fixed function IP. The functionality of the logic blocks and the routing between the blocks and memory are controlled by the configuration technology. For simplicity the switch control logic which determines whether data from Buffer 1 or Buffer 2 is sent is not covered in this example.

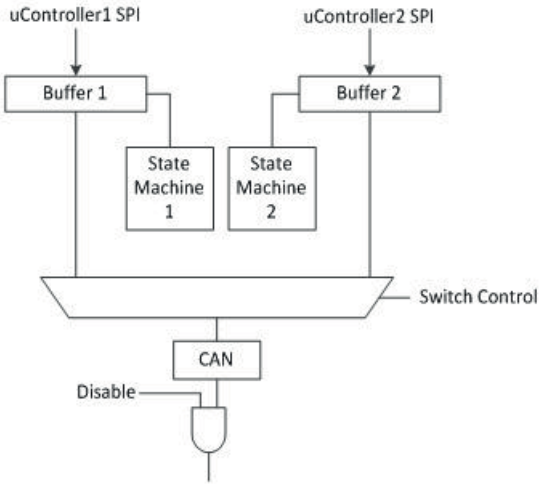

Figure 14 — PLD architecture

The design is also susceptible to intermittent and permanent hardware failures. Any chip infrastructure such as clock or power could be a source of a common mode failure. These failures can be addressed by redundancy with detection and reporting for single mode failures. Other examples include incorrect load of code at initialization and bit flip in memory. These could be detected using checksums and parity; however, some of these failures could result in a possible violation of the safety goal and would be an unacceptable risk. Error-detection-correction codes (EDC) are a superior technique as they correct errors and could report after correction that a potential problem exists in the chip. Single failures in the I/O of the chip only impact one output and would represent less risk.

NOTE 1 Depending on the functionality of the implemented circuitry it is necessary to perform further activities besides correcting the fault to restore the functionality of the design (e.g. a fault in the configuration technology leads to a non-recoverable state of a state-machine, even though the fault in the configuration technology was corrected).

If the fault has the potential to violate the safety goal without being detected by the internal safety mechanisms it would be detected by µController1 through loss of the CAN signal or a mismatch between commanded outputs and the CAN read. This is acceptable if µController1 can disable the PLD via the "Disable" signal. A dependent failure analysis is done to ensure that the risk of the PLD violating a safety goal in combination with the failure of the deactivation via the disable signal is sufficiently low.

EXAMPLE A potential hazard could occur if the switch is unable to respond to the disable command from µController1. This would be a multiple-point fault situation as if both µController1 and 2 are good; the PLD output would still represent safe values. There would not be a potential risk until one of the µControllers fails and the PLD responds incorrectly. To detect this multiple-point fault, a periodic test of the disable logic can be implemented. Since this would be performed at system level, the specific details are out of scope of this document and are not described further.

NOTE 2 In this simple example, the external measures can replace the internal safety mechanisms. In general, cases exist in which the internal measures are necessary to reach the target diagnostic coverage and therefore the detailed analysis of internal safety mechanisms described in this paragraph is applied.

Random hardware faults can be analysed by applying an inductive fault analysis (e.g. FMEA) on the design. Faults of the user design, but also faults of the PLD technology are taken into account and consider permanent and transient faults. The qualitative analysis of the design is followed up with a quantitative analysis, similar to the one described in ISO 26262-10:2012, A.3.5.

As described in 8.3.1, inputs to the quantitative analysis need to be provided by the PLD manufacturer with regard to the failure rates of the elementary parts of the PLD and the failure mode distribution.

NOTE 3 In this case of PLD internal measures, for failure mode distribution determination the approaches like described in 8.3.1.3 b) or c) are preferable.

Table 29 provides a framework for a quantitative analysis of the above design, which can be augmented with information similar to ISO 26262-10:2012, Table A.5.

NOTE 4 As discussed in ISO 26262-10:2012, A.3.3, the necessary level of detail can depend on the stage of the analysis and on the safety mechanisms used.

Table 29 — Example framework for quantitative analysis of scenario 2

Part	Sub-part	Safety related (SR) or not safety related (NSR) element?	Failure modes
I/O interface	I/O buffer	SR	Permanent
	Configuration technology	SR	Permanent
			Transient
	Routing resources	SR	Permanent
			Transient
Buffer 1	RAM data bits	SR	Permanent
			Transient
	Address decoder	SR	Permanent
			Transient
	Test/redundancy	SR	Permanent
			Transient
	Configuration technology	SR	Permanent
			Transient
	Routing resources	SR	Permanent
			Transient
Buffer 2	RAM data bits	SR	Permanent
			Transient
	Address decoder	SR	Permanent
			Transient
	Test/redundancy	SR	Permanent
			Transient
	Configuration technology	SR	Permanent
			Transient
	Routing resources	SR	Permanent
			Transient
State Machine 1	Logic blocks	SR	Permanent
			Transient
	Configuration technology	SR	Permanent
			Transient
	Routing resources	SR	Permanent
			Transient

NOTE 1 Depending on the role of each PLD part in the system, a more detailed analysis can be necessary.

NOTE 2 The example in Table 28 does not list the quantitative numbers for simplicity. An example for this can be found in 8.3.

Table 29 *(continued)*

Part	Sub-part	Safety related (SR) or not safety related (NSR) element?	Failure modes
State Machine 2	Logic blocks	SR	Permanent
			Transient
	Configuration technology	SR	Permanent
			Transient
	Routing resources	SR	Permanent
			Transient
Multiplexer	Logic blocks	SR	Permanent
			Transient
	Configuration technology	SR	Permanent
			Transient
	Routing resources	SR	Permanent
			Transient
CAN	Logic	SR	Permanent
			Transient
	RAM data bits	SR	Permanent
			Transient
	Address decoder	SR	Permanent
			Transient
NOTE 1 Depending on the role of each PLD part in the system, a more detailed analysis can be necessary.			
NOTE 2 The example in Table 28 does not list the quantitative numbers for simplicity. An example for this can be found in 8.3.			

The analysis also includes PLD related external components such as power supplies, clocks and reset circuitry. Further, if the configuration of the PLD is loaded from an external device, it is analysed if the loading of the configuration into the PLD is considered safety related or if the process of loading the configuration can lead to a failure of the item. In particular, if the PLD is loaded from µController1, common cause failures in µController1 that affect the loading mechanism and µController1 functionality is considered.

A dependent failure analysis is performed if separate channels or diagnostic measures are implemented in the PLD. An example of such an analysis can be found in ISO 26262-10:2012, A.3.6. In this example independence of the individual sub-parts is not considered as the detection of a fault of the PLD is performed by reading back the output of the CAN module with a µController.

9 Base failure rate estimation and ISO 26262 (all parts)

9.1 About base failure rate estimation

9.1.1 Impact of failure mechanisms on base failure rate estimation

The scope of this chapter is to give clarifications, guidelines and examples on how to calculate and use the base (or raw) failure rate. Base failure rate is defined as the intrinsic failure rate of an element which does not consider effects of safety mechanisms. Base failure rate is a primary input for calculation of quantitative safety metrics according to ISO 26262-5.

Quantitative safety analysis in ISO 26262 (all parts) focuses on random hardware failures and excludes systematic failures.

Each technique available for base failure rate estimation makes assumptions of failure mechanisms to be considered. Differences in results obtained from different base failure rate estimation techniques are often due to a lack of consideration for the same set of failure mechanisms amongst techniques. Results from use of different techniques applied to the same component are unlikely to be comparable without harmonization on a common set of failure mechanisms. Attention is taken with failure rate value provided without reference to the mechanisms and techniques used to derive the data. Failure mechanisms for semiconductors are dependent on circuitry type, implementation technology, and environmental factors. As semiconductor technology is rapidly evolving, it is difficult for published functional safety standards to keep pace with the state of the art, particularly for deep submicron process technologies. Because of this, it can be helpful to consider the publications of industry groups such as JEDEC (Joint Electron Device Engineering Council), ITRS (International Technology Roadmap for Semiconductors), and the SEMATECH/ISMI Reliability Council to get a broad view of semiconductor state of the art.

JEDEC, the Joint Electron Device Engineering Council, is a semiconductor industry standards organization which publishes several documents which can be helpful in providing references to understand and estimate specific failure models. JEDEC documents have the benefit of frequent updates and large scale peer review within the semiconductor industry. The following JEDEC documents can currently be accessed without fee via registration at http://www.jedec.org:

— Reference[24] summarizes many different well understood and industry accepted failure mechanisms for silicon and packaging; it can also be used to provide a physics of failure model for estimation of failure rates for the identified failure mechanisms.

— Reference[25] summarizes a number of transient fault mechanisms related to exposure to naturally occurring radiation sources and provides guidance on how to experimentally derive failure rates for susceptibility to soft error.

9.1.2 Considerations in base failure rate estimation for functional safety

ISO 26262 makes a distinction between systematic and random failures, as do other functional safety standards. Most available techniques for base failure rate estimation are intended to provide reliability estimates and make no such distinction. The user of such techniques may find that results may be excessively conservative due to inclusion of factors which estimate systematic failures. For example, estimation techniques based on observations of field failures may not have appropriate sample size or observation quality to differentiate between systematic and random failures. Similarly, models which include systematic capability as part of the base failure rate calculation may be challenging to use in an ISO 26262 context (e.g. Π_{pm} and $\Pi_{process}$ factors defined in reference[16]).

The use of the terms fault, error, and failure is done carefully. In ISO 26262 (all parts), faults create errors which may lead to a failure. In many reliability modelling standards the terms fault and failure are used interchangeably. Efforts can be made to harmonize to one set of terminology throughout the base failure rate estimation activity.

The user can also consider the modelling of distribution of failure rate over time. Many standardized models make use of a “bathtub curve” simplification, which assumes that early life (infant mortality) defects have been effectively screened by the supplier and that “wear out” (end-of-life) failure mechanisms, such as electro-migration, time –dependent dielectric breakdown, hot carriers, or negative bias temperature instability will effectively occur at negligible rates during useful mission lifetime.

Another concern with reliability standards is the handling of diagnostics which can be used to enhance availability. Consider a common SECDED ECC (Single Error Correct/Dual Error Detect Error Correcting Code) used in many state of the art automotive functional safety electronics. A reported MTTF (mean time to failure) for an SRAM with SECDED ECC may not consider a fault which results in a correctable error – thus mixing effects of base failure rate and diagnostics, which should be separated for calculation of ISO 26262 metrics.

SPFM, LFM, and failure rates used for the quantitative safety analysis like calculation of PMHF are sometimes misunderstood as a reliability prediction. A careful distinction between reliability and functional safety is necessary.

9.1.3 Techniques for base failure rate estimation

There are many different techniques which can be utilized for base failure rate estimation. In general these techniques can be summarized to fit into a few categories:

— Failure rates derived from experimental testing, such as:

— High Temp Operational Life (HTOL) testing for gate oxide breakdown.

— Reliability test chip and/or on-chip test structures to assess intrinsic reliability of the silicon technology.

— Soft error testing based on exposure to radiation sources.

— Convergence characteristic of acceleration test for screening.

— Failure rates derived from observation of field incidents, such as analysis of material returned as field failures.

— Failure rates estimated by application of industry reliability data books and/or estimated by procedural models (enhanced data books incorporating physics of failure elements), such as:

— IEC/TR 62380[23];

— SN 29500[36];

— FIDES[16].

9.1.4 Documentation on the assumptions for base failure rate calculation

When calculating the base failure rate the supplier provides a documentation describing the assumptions made and supporting rational.

EXAMPLE The selected method to calculate the failure rate (e.g. industry source or field data), how the non-operating time and solder joint were taken into account, for failure rate derived from field data which model has been used Weibull or exponential models, etc.

9.2 (General) clarifications on terms

9.2.1 Clarification of transient fault quantification

As described in ISO 26262-1, electromagnetic interference (EMI) and soft error are possible causes of transient faults. Transient faults are defined as faults that occur once and subsequently disappear.

NOTE 1 Transient faults can appear due to electromagnetic interference, which can lead to bit-flips in certain memories. Soft errors such as Single Event Upset (SEU) and Single Event Transient (SET) are transient faults.

Transient faults caused by EMI or cross-talk, even if they may lead to the same effects as other transient faults, are not quantified because they are mostly related to systematic effects that can be avoided with proper techniques and methods during design phase (e.g. cross-talk analysis during component development back-end).

Transient faults causing soft error initiated by internal or external α, β, neutron, or γ radiation sources are HW random failures that can be quantified with a probabilistic method often supported by measured data.

The ISO 26262-1:2011, definition 1.42 specifies that permanent, intermittent and transient faults (especially soft-errors) are considered. ISO 26262-5:2011, 8.4.7, NOTE 2, specifies that the transient faults are considered when shown to be relevant due, for instance, to the technology used. Therefore those faults are integrated in the safety analysis when judged applicable depending on the impact of the faults. The analyses for transient faults and permanent faults are done separately. This holds for qualitative and quantitative analyses.

Each elementary part type (flip flops, latches, memory elements, analogue devices) is investigated if it is relevant to soft error rate, specifically with respect to direct or induced alpha particles and neutrons. The relevance to those phenomena depends on the semiconductor front end technology and the materials on top of the die's surface including the package, e.g. the mould compound, the solder material (flip chip).

EXAMPLE Base failure rate for alpha particles can be influenced by the type of package, e.g. low alpha (LA) or ultra-low alpha (ULA) emitting semiconductor assembly materials.

Depending on factors such as the technology and on the operating frequency, transient fault models like single event upset (SEU), multiple-bit upset (MBU) and single event transient (SET) are considered as in references[9] and[30].

NOTE 2 Destructive single event effects like Single Event Latch-up (SEL), Single Event Burnout (SEB), and Single Event Gate Rupture (SEGR) are not considered as transient faults because these faults lead to permanent effects.

ISO 26262-10:2012, A.3.4 also describes possible sources of base failure rates for transient faults, i.e. data provided by semiconductor industries derived from JEDEC standards such as JESD 89A[25] or the International Technology Roadmap for Semiconductor (ITRS).

JESD 89A[18] is considered as the main reference related to measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductors. In that context, the base failure rate for soft errors is provided together with the conditions in which it has been computed or measured.

NOTE 3 Conditions such as neutron particle flux, altitude, temperature, and supply voltage are relevant to transient failure rate estimation of soft errors. JESD 89A[18] is used to understand those conditions.

ISO 26262-5:2011, 9.4.2.3, NOTE 5, states that situations when the item is in power-down mode shall not be included in the calculation of the average probability of failure per hour to prevent the artificial reduction of the average probability per hour. Therefore the base failure rate for soft errors is provided without derating it with respect to the operational profile of the item.

NOTE 4 The consideration of operation and non-operation time for random hardware failure rates derived from industry standards cannot be used directly for soft errors in general. A consideration of operation and non-operation time for transient faults like soft errors can be used only if a supporting rationale is available.

NOTE 5 If semiconductor provider delivers a derated soft error rate, information about the derating factor is made available for example in the Safety Manual as defined in ISO 26262-10:2012, A.3.10.

Moreover, the base failure rate for soft errors is provided without derating it with respect to architectural or application vulnerability factors.

NOTE 6 Architectural vulnerability factor (AVF) is the probability that a fault in a processor structure will result in a visible error in the final output of a program as described in reference[33].

NOTE 7 Vulnerability factors are considered in the consideration of the amount of safe faults, as described in ISO 26262-10:2012, A.3.3.2.

9.2.2 Clarification on component package failure rate

The semiconductor providers in the estimation of a hardware component failure rate take into account the failures related to the silicon die, to the enclosure/encapsulation (e.g. case) and to the connection points (e.g. pin). The connections between the connection points to the board (e.g. solder joint) are considered as board failures and are typically taken into account by the system integrator during the safety analysis at the system level.

NOTE 1 The package failure rate $\lambda_{package}$ as calculated in IEC/TR 62380:2004, 7.3.1 corresponds to the failure models inside of the package itself (including for e.g. the connection between the die and the lead frame) but it also includes the failure rate related to the connection between the package connection points and the board (solder joints).

NOTE 2 The failure rate of the hardware component calculated in SN 29500–2 includes the failure models related to the die and to the package however unlike IEC/TR 62380 it does not include the failure rate of the connection between the package connection points and the board which is treated separately in SN 29500–5.

NOTE 3 FIDES provides separate failure rates for package (cases) and solder joints due to thermal cycling.

9.2.3 Clarification on power-up and power-down times

Base failure rate is provided along with the mission profile used. If the power-up and power-down times are defined in the mission profile then they can be used to compute stress factors as foreseen by reliability handbooks like IEC/TR 62380 (τ_{on} and τ_{off}) and SN 29500 (π_w). This is shown, for permanent base failure rate, in ISO 26262-10:2012, A.3.4.2.1, using a power-down time equal to zero will provide conservative values. However power-down and power-up times are considered in order to harmonize base failure rate derived from other data sources.

9.3 Permanent base failure rate calculation methods

9.3.1 Permanent base failure rate calculation using industry sources

9.3.1.1 General

ISO 26262-5:2011, 8.4.3 states that failure rates data can be derived from a recognized industry source. This clause gives guidelines and examples on the calculation of the base failure rate for the following industry sources:

— SN 29500[36];

— IEC/TR 62380[23];

— FIDES[16].

9.3.1.2 IEC/TR 62380

A calculation method of the base failure rate for both die and package is described in IEC/TR 62380:2004, 7.3.1. Several parameters are required to determine the failure rate:

— A base failure rate per transistor per type of technology used (λ_1). A λ_1 value is provided for different type of integrated circuits families in IEC/TR 62380:2004, Table 16;

— A failure rate related to the mastering of the technology and valid for the whole component regardless of its complexity (λ_2);

— A base failure rate related to the package (λ_3);

— The number of transistors of the hardware component (N);

— The difference between the year of manufacturing or technology release/update and the reference year (1998) (α);

— The operating and non-operating phases seen by the hardware component (τ, τ_{on} and τ_{off});

— A temperature stress factor (π_t) applicable to the die part of the component;

— The number and the amplitude of the temperature cycling seen by the hardware component (n_i and ΔT_i);

— The mismatch between the thermal coefficients of the board and the package material (α_S and α_C);

NOTE In IEC/TR 62380:2004, Table 16, “actual number” corresponds to the real number of transistors regardless the sizes of those transistors.

9.3.1.2.1 Die base failure rate calculation using IEC/TR 62380

Multiple interpretations exist about how to combine λ_1 and λ_2 in case of circuit elements with different technologies (CPU, memories, etc.) implemented in the same device.

In ISO 26262-10:2012, A.3.4.2.2, each circuit element inherits the λ_1 and λ_2 of the respective technologies, so basically the λ_1 and λ_2 are cumulated. As described in ISO 26262-10:2012, A.3.4.2.2, Note 2, to simplify calculation, estimation can be done using a single selection of λ_1 and λ_2 for the entire device. In particular, this is the case if the user can identify a perfect match between its product and one of the integrated circuits families type listed in IEC/TR 62380:2004, Table 16 then the user can directly apply the failure rate calculation method as described in IEC/TR 62380:2004, 7.3.1.

As alternative approach to the ones described in ISO 26262-10:2012, A.3.4.2.2, if the die of the hardware component is composed of different type of elements for which no matching technology can be identified in IEC/TR 62380:2004, Table 16 (for e.g. BICMOS chips with both, linear circuits < 6V and digital circuits) then the die base failure rate (λ_{die}) can be calculated by summing the products of $\lambda_{1,element}$ and $N_{element}$ values corresponding to the different matching types and using a single (conservative) maximum λ_2 value:

$$\lambda_{die} = \left(\left\{ \sum_{elements} \left(\lambda_{1,element} \times N_{element} \right) \times e^{-0,35 \times a} + Max(\lambda_{2,element}) \right\} \times \left\{ \frac{\sum_{i=1}^{y} \left(\pi_t \right)_i \times \tau_i}{\tau_{on} + \tau_{off}} \right\} \right) \quad (1)$$

Two examples are given below to illustrate both situations (i.e. matching device type identified or not):

Table 30 — Microcontroller example with matching device type

Circuit Element	λ_1 FIT	N (gate or transistors)	α	λ_2 FIT	Base failure rate FIT	De-rating for temp	Effective failure rate FIT
50k gate CPU	$3,4 \times 10^{-6}$	200 000 (4 transistors/gate)	10	1,7	1,73	0,17	0,29
16kB SRAM		786 432 (6 transistors/bit for a low-consumption SRAM)					
Die failure rate (FIT)							0,29
Die failure rate (FIT, τ_{off} sets to zero)							0,84

NOTE 1 In case τ_{off} time is set to zero then the die failure rate of the microcontroller example given above is 0,84 FIT as described in ISO 26262-10:2012, A.3.4.2.2.

Table 31 — Mixed signal example without matching device type

Circuit Element	λ_1 FIT	N (gate or transistors)	α	Base failure rate without λ_2 FIT	λ_2 FIT	De-rating for temp	Effective failure rate FIT
Digital Circuits	$1{,}0 \times 10^{-6}$	28 000	10	0,00085	1,7		
Linear/digital circuits low voltage (<6V)	$2{,}7 \times 10^{-4}$	30 000		0,25	20		
Die failure rate (FIT)				0,25	Max(20,1,7) = 20	0,17	3,4

NOTE 2 In the table above, two circuits (one digital and one linear/digital low voltage) are considered with the respective value of λ_1 and λ_2. The maximum value of λ_2 is selected.

9.3.1.2.2 Package base failure rate calculation using IEC/TR 62380

The package failure rate $\lambda_{package}$ as calculated in IEC/TR 62380:2004, 7.3.1 corresponds to the failure modes inside of the package itself (including for e.g. the connection between the die and the lead frame) but it also includes the failure rate related to the connection between the package connection points and the board (solder joints).

Table 32 — Package base failure rate calculation example

Package type	ΔT_j °C	S (Number of pins)	D mm	Π_α	λ_3 FIT	De-rating for temperature cycling	Effective failure rate FIT
PQFP 144	26,27	144	26,58	1,05	11,87	6027	207
Package failure rate including solder joints between package and board (FIT)							207
Total package failure rate without solder joints between package and board (FIT)							16

NOTE 1 The package in the example is a 144 pin quad flat package and cooled by natural convection. The power consumption is 0,5 W leading to an increase of the junction temperature ΔT_j of 26,27°C. The value of D and λ_3 are computed using the Table 17b in reference[23] on the basis of the following values: pitch = 0,5 mm and width = 20 mm.

NOTE 2 In the case λ_3 value provided by IEC/TR 62380 is not suitable, the supplier of the component can replace this value with supplier's internal base failure rate or with a more up to date value from other industry sources. In such case an argument is provided by the supplier to justify the value of the base package failure rate that has been used.

9.3.1.3 SN 29500

The SN 29500 follows a table look up approach. Expected values for failure rates under specified reference conditions are given. Values are to be looked up in tables using product type, technology and transistor count as an input. If the integrated circuits are operated under conditions different from the reference conditions a calculation from reference to operating conditions is to be used. The calculation takes into consideration temperature, voltage and drift (for analogue elements). For the temperature part of the calculation to operating conditions a modified Arrhenius equation is used.

Parameters required for the calculation of the failure rate with SN 29500:

— N, the number of equivalent transistors;

— λ_{ref}, the basic failure rate for the hardware component, based on the process technology;

— ΔT_j, the junction temperature increase;

— The mission profile of the hardware component.

NOTE In the case the number of equivalent transistors N is not listed in the failure rates families Tables 1, 2 or 3 of SN 29500–2:2010 and when possible the user can use for example a log linear interpolation to determine the equivalent λ_{ref} value and a logarithmic interpolation to determine $\theta_{vj,1}$ (virtual junction temperature).

EXAMPLE For "microprocessors and peripherals, microcontrollers and signal processors" family as defined in SN 29500–2:2010, Table 2, the following interpolation example is done to determine λ_{ref} and $\theta_{vj,1}$ values.

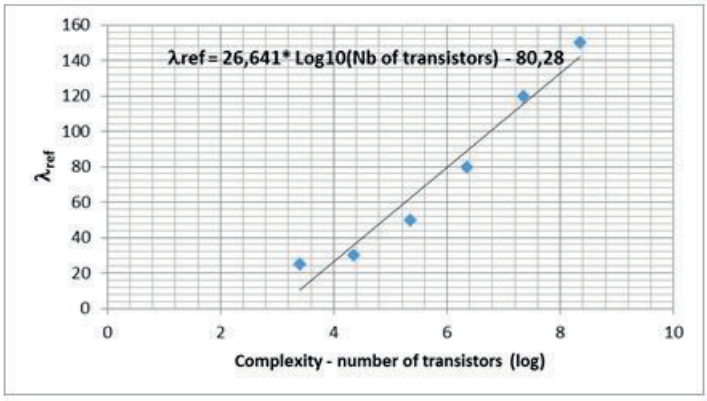

Figure 15 — Log linear interpolation of λ_{ref}

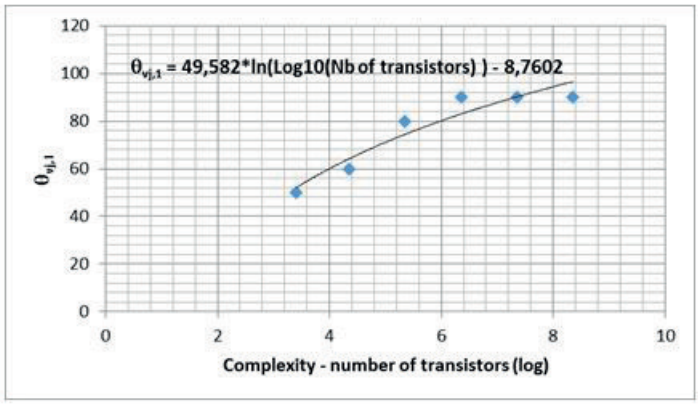

Figure 16 — Logarithmic interpolation of the virtual junction temperature $\theta_{vj,1}$

So for a microcontroller which has for example 650 million transistors the corresponding λ_{ref} and $\theta_{vj,1}$ values would be respectively 154,50 FIT and 99,14°C.

9.3.1.3.1 Failure rate calculation for the microcontroller example without non-operating phase

For the microcontroller example of ISO 26262-10:2012, A.3.4.2.1 in CMOS technology with 500 k to 5 million transistors we get 80 FIT at 90 °C reference temperature condition.

Table 33 — Parameters required for failure rate calculation example with SN 29500

N **(transistors)**	**Technology and family**	λ_{ref} FIT	ΔT_j °C	**Temperature dependent reference,** Z_{ref} 1/eV	*A*	*E*a1 eV	*E*a2 eV
986 432(Digital + SRAM)	CMOS, microprocessor	80	26,27	5,11	0,9	0,3	0,7

Assuming 500 working hours per year and using the motor control mission profile as defined in IEC/TR 62380[23], we have the following result:

Table 34 — Microcontroller failure rate calculation example with SN 29500

Ambient temperature θ_U °C	**Working time** h	**Junction temperature** $\theta_{j,2}$ °C	**Dependence factor** Z 1/eV	**Temperature dependence factor** $\Pi_T(\theta_U)$
32	172,4	58,27	2,04	0,27
60	129,3	86,27	4,77	0,85
85	198,3	111,27	6,87	2,51
Overall Temperature Dependent Factor, Π_T				1,31
Effective failure rate for the overall hardware component (FIT)				104,65

9.3.1.3.2 Failure rate calculation for the microcontroller example with non-operating phase

There is a difference between IEC/TR 62380 and SN 29500 in the way the non-operating phases are considered. In IEC/TR 62380 the non-operating hours are by default included in the mission profile of the product whereas in SN 29500 only the operating hours are by default considered. In ISO 26262-10:2012, A.3.4.2.2, an alternative approach for calculating failure rate with IEC/TR 62380 was proposed by setting τ_{off} time to zero - this approach is considered to be closer to SN 29500 method.

In a similar way, operating and non-operating phases can also be taken into account in SN 29500 for the calculation of the failure rate. This is done by applying a stress factor π_ω described in SN 29500-2, Clause 4.4. Using the motor control mission profile as defined in IEC/TR 62380 and an average temperature of 10,5 °C gives a stress factor value of 0,06. Applying the calculated stress factor to the microcontroller example failure rate gives:

Table 35 — SN 29500 failure rate calculation with or without non-operating phases

N **(transistors)**	**Technology and Family**	λ_{ref} FIT	λ **Without non-operating phase** FIT	**Stress Factor**	λ **With non-operating phase** FIT
986 432 (Digital + SRAM)	CMOS, microprocessor	80	104,65	0,06	6,28

NOTE The non-operating average temperature is obtained from the average worldwide night and day-light temperatures (respectively 5 °C and 15 °C) as defined in IEC/TR 62380 and considering a 50 % ratio between night and day.

9.3.1.3.3 Method to split SN 29500 overall failure rate into die and package failure rates

As stated by the maintainer of SN 29500 the base failure rate value calculated with SN 29500 is valid for the whole hardware component only and does not provide a method to split between package failure rate and die failure rate. An estimation of the split of package and die failure rates from an SN 29500 base failure rate could be calculated by using other industry sources which provide such data or from field data statistics when available. The IEC/TR 62380 and FIDES Guide are possible industry sources for this data.

9.3.1.4 FIDES Guide

The following is an example of the estimation of hardware failure rate as needed to support quantitative analysis using the methods detailed in the FIDES guide.[16] The failure rate model for a semiconductor per FIDES guide considers the failure rate of the device to be a factor of:

— Physical contributions ($\lambda_{\text{Physical}}$);

— Process contributions (Π_{Process});

— Part Manufacturing contributions (Π_{PM}).

The first is an additive construction term comprising physical and technological contributing factors to reliability. The second is a multiplicative term including the quality and technical control over the development, manufacturing and the usage process for the product containing the device. The third factor represents for example the quality of the manufacturing site and the experience of the supplier. Π_{Process} and Π_{PM} are set to 1 as these factors are related to systematic issues.

The physical contribution is composed of stresses acceleration factors due to usage conditions and an induced (i.e. unexpected overstress) multiplicative term inherent to the application of the product containing the device.

The models used in the FIDES guide for integrated circuits include the following physical stress families:

— thermal;

— temperature cycling;

— mechanical;

— humidity.

To compute the microcontroller die and package base failure rates (i.e. before application of de-rating for operating conditions), it is necessary to consider the following elements:

— $\lambda 0_{\text{TH}}$, the basic failure rate associated with the type of device and process technology;

— physical stress parameters a and b associated with the type of package.

Those factors are combined using FIDES. Selection of parameters can be done based on the process technology, type of circuitry and package utilized by the design. Values are available related to Microprocessor, Microcontroller, DSP and SRAM, and PQFP package with 144 pins.

Table 36 and Table 37 below show the computation of the failure rates used in the quantitative example of a CMOS technology based MCU which consumes 0,5 W power. The microcontroller die is packaged in a 144 pin quad flat package and cooled by natural convection and low-conductivity board.

Table 36 — Base failure rate of the die from UTE FIDES

Circuit element	$\lambda 0 T_{\text{H}}$ FIT
50 k gate CPU	0,075
16 kB SRAM	0,055
Sum	0,13

Table 37 — Base failure rate of the package from UTE FIDES

Package	$\lambda 0T_{Cy\_Case}$			$\lambda 0T_{Cy\_Solderjoints}$		
	a	b	$\lambda 0T_{Cy\_Case}$ FIT	a	b	$\lambda 0T_{Cy\_Solcerjoints}$ FIT
144 pin PQFP	12,41	1,46	0,0058	10,80	1,46	0,029

Once the base failure rate for the microcontroller die and package has been generated, a derating factor is applied based on thermal effects and operating time. The derating factor takes into account:

— Junction temperature of the microcontroller die, which is calculated based on:

  — power consumption of the microcontroller die;

  — package thermal resistance, based on package type, number of package pins and airflow;

— An application profile which defines 1 to Y usage phases, each of which is composed of an application "on-time", "cycle time", "cycle delta temperature", and "cycle max temperature", and "ambient temperature".

It is noted that the profile for use in the model considers more/other parameters than those provided in the profile of reference[23].

At first, the simplified mission profile example shown in Table 38 is considered.

Table 38 — Simplified mission profile example

			Thermal	Thermal cycling			
PHASE	**On/Off**	$t_{annual\text{-}phase}$ h	$T_{ambient}$ °C	$\Delta T_{cycling}$ °C	θ_{cy} h	$N_{cy\text{-}annual}$ h	$T_{max\text{-}cycling}$ °C
non-operational day	Off	720	15	10	24,0	30	20
night start	On	168	60	55	0,25	670	60
day start	On	335	60	45	0,25	1340	60
off - operational day	Off	7,538	15	10	22,5	30	20

The die base failure rate with derating factors is as follows:

Table 39 — Die base failure rate with temperature derating factor

Circuit element	$\lambda 0T_H$ FIT	Derating for temperature	Effective failure rate FIT
50k gate CPU	0,075	5,79	0,43
16kB SRAM	0,055	5,79	0,32
Sum	0,13		0,75

For evaluating these derating factors, the junction temperature, i.e. ΔT_j due to self-heating is calculated as 18K, using the parameters and formula described in FIDES.

Table 40 — Package base failure rate with temperature cycling derating factor

	λT_{Cy_case}			$\lambda T_{Cy\_solderjoints}$		
Package	$\lambda 0 T_{Cy\_case}$ FIT	**Derating for cycling**	**Effective failure rate (FIT)**	$\lambda 0 T_{Cy\_solderjoints}$ FIT	**Derating for cycling**	**Effective failure rate** FIT
144 pin PQFP	0,0058	130	0,75	0,029	10	0,28

Then, the elaborated mission profile example shown in Table 41 is considered.

Table 41 — Elaborated mission profile example

			Thermal	**Thermal cycling**			
PHASE	**On/Off**	$t_{annual\text{-}phase}$ h	$T_{ambient}$ °C	$\Delta T_{cycling}$ °C	θ_{cy} h	$N_{cy\text{-}annual}$ h	$T_{max\text{-}cycling}$ °C
non-operational day	Off	720	14	10	24,0	30	19
night start	On	117	32	22	0,0	670	32
day start	On	58	32	18	0,0	1340	32
full load operation	On	201	85	53	1,0	335	85
highway operation	On	131	60	28	4,0	30	60
off - operational day	Off	7,532	14	10	23,0	30	19

The derating factors are as follows:

Table 42 — Effective failure rate

Circuit element	$\lambda 0 T_H$ FIT	**Derating for temperature**	**Effective failure rate** FIT
50k gate CPU	0,075	12,44	0,93
16kB SRAM	0,055	12,44	0,68
Sum (FIT)	0,13		1,61

For evaluating these derating factors, the junction temperature, i.e. ΔT_j due to self-heating is calculated as 18K, using the parameters and formula described in FIDES.

Table 43 — Package and solder joints failure rate

	λT_{Cy_case} FIT			$\lambda T_{Cy\_solderjoints}$ FIT		
Package	$\lambda 0 T_{Cy\_case}$ FIT	**Derating for cycling**	**Effective failure rate** FIT	$\lambda 0 T_{Cy\_solderjoints}$ FIT	**Derating for cycling**	**Effective failure rate** FIT
144 pin PQFP	0,0058	42	0,25	0,029	4	0,12

The component package failure rate is then 0,25 FIT. The solder joints failure rate value in Table 43 is given as information only and is not considered as part of the package failure rate.

9.3.2 Permanent base failure rate calculation using field data statistics

The following section is about the failure rate estimation using field data statistics. As it is very difficult to get an appropriate estimation, field data statistics are only used with special care. A thorough

analysis of the field return process is performed and the result of the analysis is used for the quantitative evaluations. In particular the following topics are evaluated:

— How does the field return process handle known quality issues?

— What kind of information is available about the real mission profile?

— What is the effectiveness of the field monitoring process?

Because the methodology used to calculate the failure rate from field data has an influence on the confidence level of the resulting failure rate value, the following points are taken into account by the semiconductor suppliers:

— A proper field data collection system needs to be put in place in accordance with ISO 26262-2:2011, 7.4.2.4;

— The goal of the used method is not to approximate as close as possible the real failure rate but to provide a failure rate value for which there is a high confidence that it is above the real failure rate value.

— Depending on the quantity and the quality of the field return and on which part of the bathtub curve we want to model, an exponential model method (9.3.2.1) can be used. The failure rate during the useful life of the product can be considered as constant and therefore estimated by the exponential model;

— A possible approach is that only the failures occurring during the warranty period of the car are considered. Over the useful life of the product, the failure rate can be considered as constant;

— Only random hardware faults are considered (for e.g. not considering systematic faults due to process issue, EOS or weak design, etc.);

— Because not all failures in the field may reach the semiconductor suppliers, a correction factor can be applied to the total number of returns. The correction factor may depend on the application and the device population used to estimate the field based failure rate. A typical value for the correction factor is between 5 and 10, but it can be more. This number is documented and agreed between the supplier and the customer. After the warranty period a correction factor cannot be determined since after this period the failures from the field might not be returned anymore and therefore not analysed.

— An acceleration factor AF corresponding to the temperature stress or to the thermal cycling stress effects can be respectively calculated using Arrhenius model associated with a specified activation energy or Coffin-Manson equation;

— The total operating time of the products in the field can be estimated using the mission profiles of the products when available. If not then typical average yearly operating time of 500 h can be used in combination with a standard mission profile as defined for example in IEC/TR 62380 or provided by any other organizations or industry standard. The variability in car usage from the drivers can also be taken into account by estimating the quantity of hours spent in field using for example a mean of 500 h a year with a standard deviation of 145 h;

— The mission profile of the filed data are documented and considered appropriately in the quantitative evaluations;

— Systematic failures are only removed from the field statistics if the source of the systematic failure has been mitigated.

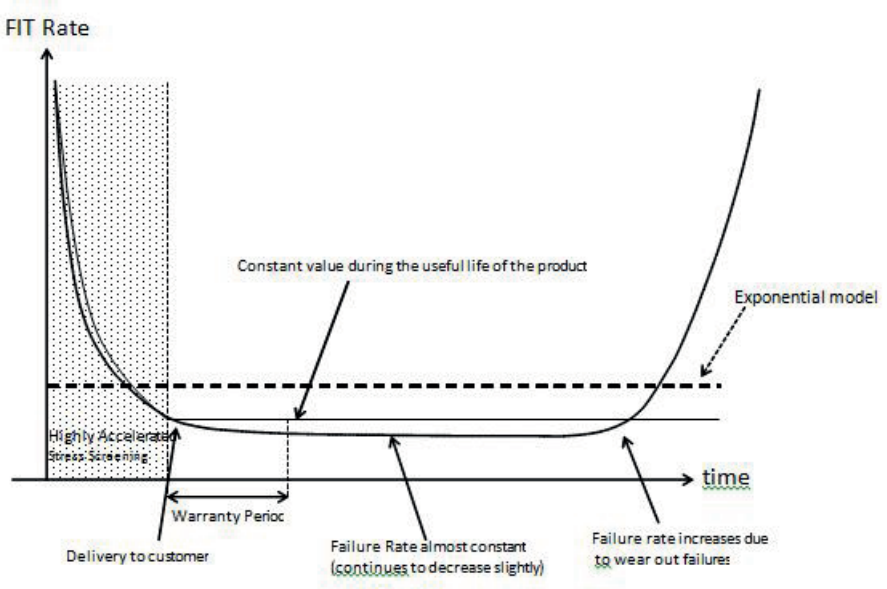

Figure 17 — Bathtub curve – evolution of failure rate over time

9.3.2.1 Exponential model method

The exponential model can be used in general to determine a constant failure rate from field returns. In the special case of a small number of field returns the exponential model is recommended to be used instead of the Weibull model method. By using the exponential model we consider that the failure rate is flat over time. Exponential model is suitable to predict the failure rate during the useful life of the product where the failure rate can be considered constant. In this case of a constant failure rate, χ^2 (chi-squared) statistical function gives a good approximation of the failure rate. It is proposed to use for example an interval estimator with a one-sided upper interval estimation at 70 % confidence level instead of using a point estimator for the failure rate. That means that with 70 % probability, the real value of the failure rate is below that value. The failure rate can be calculated using the formula below:

$$\text{FIT} = \frac{\chi^2_{\text{CL},2n+2} \times 10^9}{2 \times \text{cumulative operational hours} \times f_a}$$

where

n is the number of failures multiplied by the correction factor (f_c);

CL is the confidence level value (typically 70 %);

f_a is the acceleration factor.

NOTE The acceleration factor is used to adapt failure rate values from one mission profile to another one as described in 9.3.3.

9.3.3 Calculation example of hardware component failure rate

In this clause an example of a die failure rate calculation using field data statistics is given using the exponential model method. The numbers used are arbitrarily chosen and shall be replaced by real data.

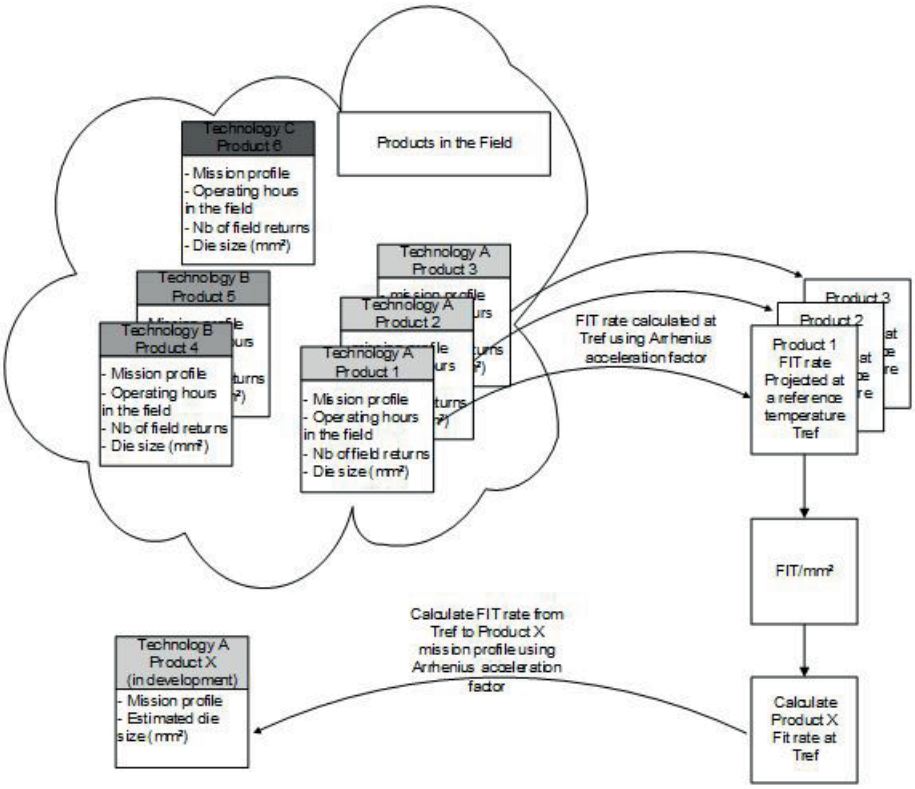

Figure 18 — Die failure rate calculation method using field data statistics

In this example we assume that the semiconductor supplier is collecting statistics from 3 products in the field as described in table below:

Table 44 — Mission profile and equivalent junction temperature, $T_{j,eq}$

Tj °C	**Chip 1** **Phase Duration** h	Tj °C	**Chip 2** **Phase Duration** h	Tj °C	**Chip 3** **Phase Duration** h
–20	1 000	–25	100	–20	500
10	2 000	10	500	15	800
30	1 500	35	10 000	45	6 000
45	6 000	55	8 000	80	4 200
70°C	1 000	90°C	1 000	100°C	600
100°C	1 300	100°C	200	120°C	300
130°C	200	120°C	200	150°C	100
Mission profile Equiv. Temp, $T_{j,eq}$	**55,1°C**	**Mission profile Equiv. Temp, $T_{j,eq}$**	**51,4°C**	**Mission profile Equiv. Temp, $T_{j,eq}$**	**67,4°C**
Total duration	13 000	**Total duration**	20 000	**Total duration**	12 500

NOTE 1 The mission profile equivalent temperature $T_{j,eq}$ corresponds to the temperature that would have the same effect as the whole mission profile from a temperature stress perspective. $T_{j,eq}$ can be calculated using the Arrhenius equation. In the above example an activation energy E_a of 0,3 eV was assumed.

Table 45 — Calculation of failure rate per mm² at reference temperature, T_{ref}

Product Name	Die size mm²	Mission profile equivalent temp, $T_{j,eq}$ °C	Total Device Operating hours in million device hours	Arrhenius Acceleration Factor	Equivalent Operating hours at a T_{ref} of 55°C in million device hours	Equivalent die area hours at a T_{ref} of 55°C in million mm² hours	Number of failures during warranty period	Number of Failures with a Correction factor of 5
Chip1	30	55,1	7 000	1,00	7 022,67	210 680	1	5
Chip2	25	51,4	10 200	0,89	9 066,96	226 674	1	5
Chip3	50	67,4	5 000	1,47	7 359,25	367 963	2	10
Total die area hours						805 317	**Total number of failures**	20
FIT/mm² **at T_{ref} of 55°C**						0,029		

NOTE 2 The device operating hours of the different devices can be summed up together if they are referred to the same reference temperature T_{ref}. In this example T_{ref} is 55°C and the equivalent devices hours at T_{ref} are calculated using Arrhenius equation associated with an activation energy E_a of 0,3 eV.

NOTE 3 The failure rate per mm² value at the reference temperature T_{ref} is calculated using the χ^2 statistical function from the total number of failures and the total number of die area hours. In this example an upper confidence level of 70 % has been used.

As explained in Figure 18, the failure rate per mm² at T_{ref} derived from the field data statistics can then be used to calculate the failure rate of the target product under design:

Table 46 — Final chip failure rate calculation

	Mission profile Equiv. Temp, $T_{j,eq}$ °C	Die size mm²	FIT/mm² at T_{ref}	Arrhenius Acceleration Factor	FIT/mm² at Equiv. Temp ,$T_{j,eq}$	Die base failure rate FIT
Target Chip under design	75	23	0,029	1,84	0,053	1,22

NOTE 4 Same method is applied to calculate package failure rate but the acceleration factor is calculated using Coffin-Manson or Norris-Landzberg model (as discussed in reference,[22] subclause 5.2.7.10 “Failure Modes”, reference,[24] Chapter 5.14 and reference,[16] Chapter 2.5.1 “Physics of failures and models”) instead of Arrhenius model. Figure 19 gives an overview of the methods used to calculate the package failure rate using field data statistics.

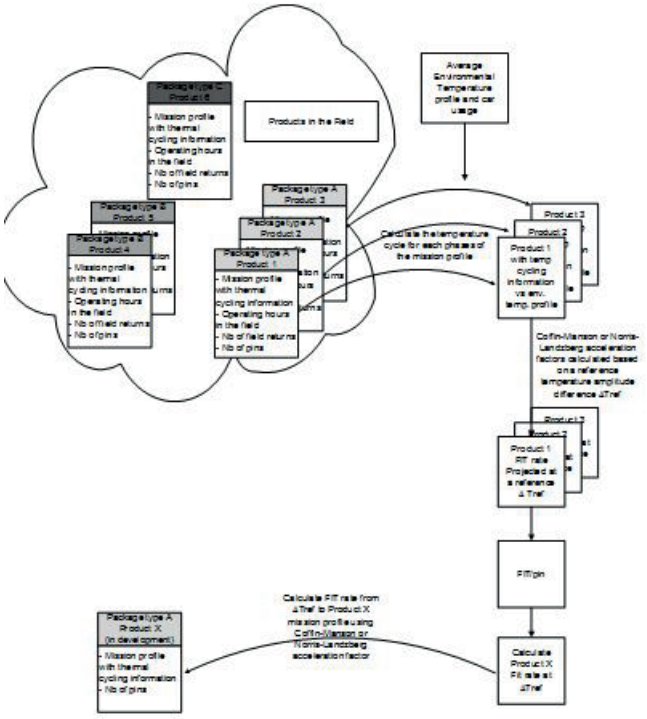

Figure 19 — Package failure rate calculation method using field data statistics

NOTE 5 In case no distinction is done in the field data analysis between die and package (as it is the case for example in SN 29500) then Arrhenius law can be used to calculate the hardware component (die and package) failure rate using the mission profile temperatures and reference temperature T_{ref} as depicted in Figure 18.

9.3.4 Base failure rate calculation using accelerated life tests

To de-rate from the temperature at which the life test is carried out to the maximum operating temperature an acceleration factor is applied. This calculation uses the Arrhenius equation with typical activation energy of 0,7 eV.

The number of faults obtained from the sample is used in the χ^2 distribution function with a certain confidence level to obtain the number of faults that would occur over the entire population tested.

Voltage acceleration is also taken into account when determining the life of devices. This is calculated by taking the oxide thickness into consideration and de-rating from the stress test voltage to the life operating voltage.

$$f_{aV} = \exp(\beta) \cdot [V_{test} - V_{op}]$$

where

f_{aV} is the voltage acceleration factor;

V_{op} is the gate voltage under typical operating conditions (in Volts);

V_{test} is the gate voltage under accelerated test conditions (in Volts);

β Is the voltage acceleration coefficient (in 1/Volts).

9.3.5 Failure rate distribution methods

The previous section details several methods to determine the base failure rate for the hardware component. As described the overall hardware component failure rate can be available as a single value or split into package failure rate and die failure rate (either directly from the failure rate source used or split using the optional method described previously). During the safety analysis the total failure rate is allocated to the different elements composing the hardware component or to the different failure modes related with those elements.

Different distribution methods can be applied (see also ISO 26262-10:2012, A.3.3.1 and A.3.4.2.3):

— Failure rate distribution to the die part: failure rate for internal elements of the component (like for example digital blocks, analogues blocks and memories): two methods can be considered to perform the distribution:

  — The first method consists of using a failure rate per mm^2 value obtained by dividing the die failure rate or the whole hardware component failure rate (if not separated into package and die contributions) by the die area of the hardware component. The failure rate distribution is done by multiplying the part or sub-part area related to the failure mode under analysis by the failure rate per mm^2 value.

  — The second method is based on base failure rates and elementary parts. This is done by making an estimation of the number of equivalent gates (or number of transistors) for each part, sub-part or basic/elementary sub-part related to the failure mode under analysis.

— Failure rate distribution to the package: This can be derived only when the failure rate of a component is split between its package and die contributions. In such a case, for pins that are safety related, the distribution of the failure rate can be done using a failure rate per pin value which is obtained by dividing the package failure rate by the total number of pins of the package (safety related or not).

NOTE 1 The selection of the method used can be based on the layout (or planned layout) of the circuit under analysis or on the analysis of how failure modes are shared between the HW elements.

EXAMPLE The area based method could be more suitable when the logic related to the failure mode under analysis is located in a single region not shared with other failure modes. The base failure rate and elementary part method are applicable when the logic is spread across a larger area or if the logic is shared between more failure modes.

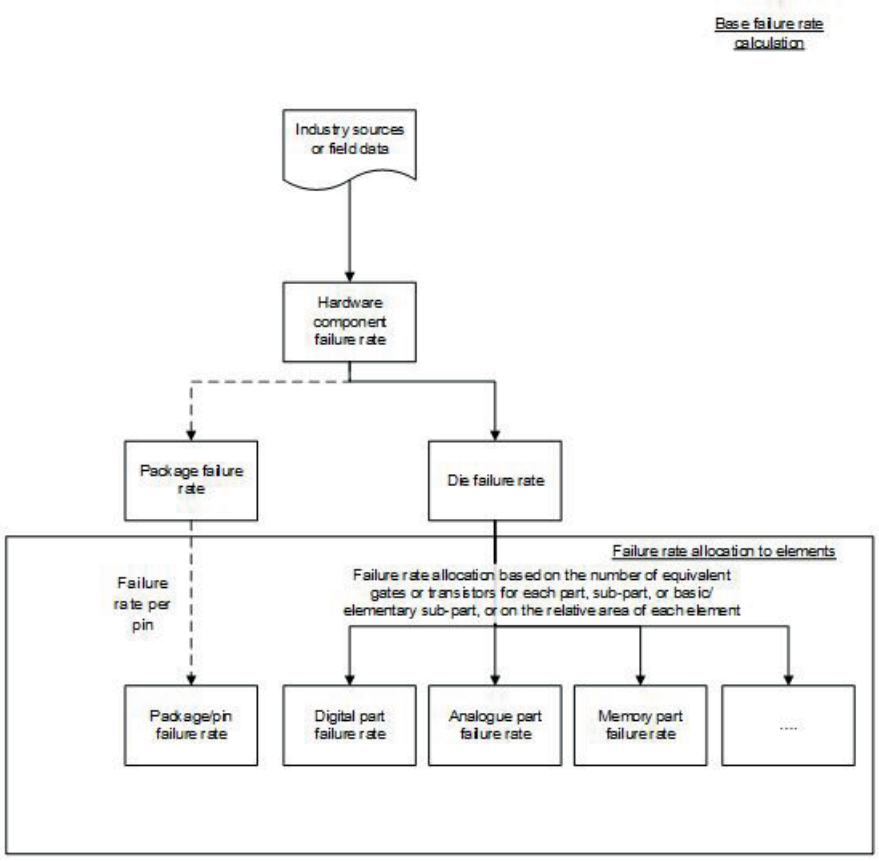

Figure 20 — Failure rate distribution

10 Semiconductor dependent failure analysis and ISO 26262

10.1 Introduction to DFA for semiconductors

The goal of this chapter is to provide guidelines for the identification and analysis of possible common cause and cascading failures between given elements, the assessment of their risk of violating a safety goal (or derived safety requirements), and, the definition of safety measures to mitigate such risk if necessary. This is done to evaluate potential safety concept weaknesses (ISO 26262-9:2011, 8.4.9) and to show the fulfilment of requirements concerning independence or freedom from interference (ISO 26262-9:2011, 7.1).

The scope of this chapter is the DFA between hardware elements implemented within one silicon die and between hardware and software elements. The elements under consideration are typically hardware-elements and their safety mechanisms (specified during the activities of ISO 26262-5).

The scope, analysis method(s) and the necessary safety measures can depend on the nature of the given elements (e.g. just SW elements, just HW elements or a mix of HW and SW elements) and the nature of the involved safety requirements (e.g. fail safe, fail operational).

A list of dependent failure root causes, from here on named “DFI” (dependent failure initiator), is provided as a starting point, considering different systematic, environmental and random hardware issues. Some random hardware DFI, e.g. shared resources or interfering elements[1)] of the elements under consideration, can be considered within the standard safety analysis once the dependencies

1) Interfering elements have the capability to corrupt resources of other hardware elements as a consequence of a random hardware fault or systematic fault: e.g. a DMA (direct memory access peripheral) writes to a wrong address and silently corrupts safety-related data.

are identified and can be classified as either residual faults, single-point faults or multiple-point faults (ISO 26262-5:2011, 9.4.2.3, Note 1). The DFA addresses those DFI, which are not addressable within the standard safety analysis, in a qualitative way.

The list of DFI also contains some typical safety measures used to address these. The necessary safety measures may depend on the nature of the safety requirement, in particular if the risk of occurrence of the dependent failure in the field is to be minimized or if it is sufficient that if the dependent failure occurs the safety goal is not violated.

The requirements that aim at controlling dependent failures need to precisely identify in which manner the control measure is intended to operate:

— In case of a fail-safe requirement of the given elements it is not necessary to avoid the occurrence of the dependent failure. It is sufficient to detect it and switch the element into a safe state, e.g. by deactivation of safety critical outputs or by reporting the error to another element that can take measures to bring the system into a safe state.

— In case of a fail-operational requirement, where deactivating the given elements might not be acceptable and no safe state can be defined that does not require at least an operation with degraded performance, safety measures might be necessary which reduce the probability of the dependent failure occurring in the field.

10.2 Relationship between DFA and safety analysis

The correlated elements for which a DFA is relevant, can already be identified from the safety analyses done in accordance to ISO 26262-5:2011, 7.4.3. These can be dual-point failure scenarios like:

— Functions and their safety mechanisms (including the fault reaction path - the chain of elements and/or tasks that are required to implement the fault reaction);

— Functional redundancies (e.g. two current drivers or two A/D converters).

And single-point (residual) failure scenarios of shared elements that belong to the semiconductor infrastructure like:

— Clock generation;

— Embedded voltage regulators;

— Any shared hardware resource used by the aforementioned correlated elements.

While the safety analysis primarily focuses on identifying single point faults and dual/multiple-point faults to evaluate the targets for the ISO 26262 metrics and define safety mechanisms to improve the metrics if required, the DFA complements the analysis by ensuring that the effectiveness of the safety mechanisms is not affected by dependent failure initiators. As mentioned in ISO 26262-5:2011, 7.4.3.1, the safety analysis can be used in a first place to support the specification of the hardware design and subsequently can be used for the verification of the hardware design. Similarly the DFA can be applied as well during the specification of the hardware design (e.g. to specify safety mechanisms for the shared elements that have been identified) and in the second stage to verify that the assumption taken during the specification are realized and reach the intended effectiveness.

10.3 Dependent failure scenarios

In Figure 21, Element A and Element B are correlated elements that have the potential to fail under the presence of an external root cause. The root cause can be related to a random hardware fault or to a systematic fault.

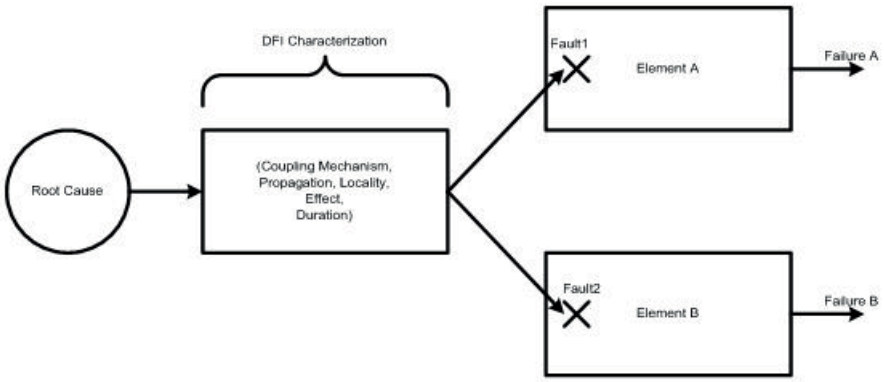

Figure 21 — Dependent failure root cause

Typical situations related to a random hardware fault can include failure of shared resources or single physical root cause. For these situations a failure rate can be quantified and needs to be considered into the safety analysis according to ISO 26262-5.

Typical situations related to systematic faults can include environmental faults, development faults, etc. For these situations it is in general not possible to make a quantitative analysis. Additionally the root cause can be located inside the semiconductor element under consideration or located outside and propagates inside the semiconductor element through signal or power supply interfaces for instance.

Figure 22 refers to coupling mechanism that aims at characterizing some exemplary properties of the disturbances created by a given root cause. Such properties can help to specify the mitigation measures and as well to define the adequate models that can be used to verify the effectiveness of the mitigation measures (see 10.5.2). They are now introduced:

— Coupling mechanism: this property characterizes the means by which a root cause induces a disturbance. Known coupling mechanisms are: Conductive coupling occurs when the coupling path between the source and the receptor is formed by direct contact with a conducting body, for example a transmission line, wire, cable, PCB trace or metal enclosure.

  — Near field coupling occurs where the source and receiver are separated by a short distance (typically less than a wavelength). Strictly, “Near field coupling” can be of two kinds, electrical induction and magnetic induction. It is common to refer to electrical induction as capacitive coupling, and to magnetic induction as inductive coupling.

    — Capacitive coupling occurs when a varying electrical field exists between two adjacent conductors typically less than a wavelength apart, inducing a change in voltage across the gap.

    — Inductive coupling or magnetic coupling occurs when a varying magnetic field exists between two parallel conductors typically less than a wavelength apart, inducing a change in voltage along the receiving conductor.

    — Radiative coupling or electromagnetic coupling occurs when source and receiver are separated by a large distance, typically more than a wavelength. Source and receiver act as radio antennas: the source emits or radiates an electromagnetic wave which propagates across the open space in between and is picked up or received by the receiver.

  — Propagation medium: this property characterizes the coupling path the disturbance uses through the semiconductor element. Typically it can be:

    — Signal lines;

    — Power supply network;

  - Substrate;
  - Package;
  - Air;
- Locality: this property characterizes if the disturbance has the potential to affect multiple elements or is limited to a single element. In the latter case the affected element is assumed to produce a wrong output that propagates to multiple elements connected to it (cascading effect).
- Effect: this property characterizes in which manner the hardware is affected by the disturbance. Possible examples are:
  - Timing violation (e.g. caused by crosstalk, timing fault, etc.)
  - Incorrect logical behaviour (e.g. caused by latch-up, etc.)
- Timing: this property characterizes some properties of the disturbance related to its propagation delay (e.g. for propagation of temperature gradient) or its timing behaviour like periodicity (e.g. in case of ripple noise over power supply), etc...

In order to illustrate the aforementioned properties two examples are given.

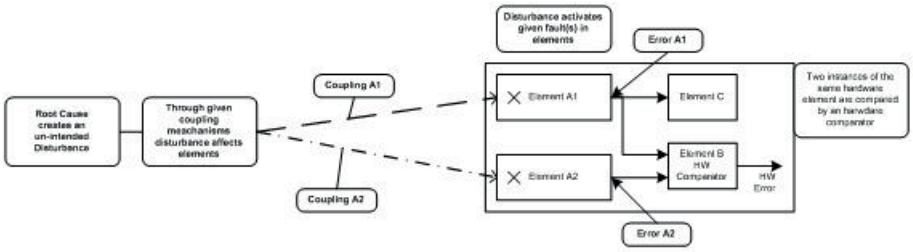

Figure 22 — Dependent failure by physical coupling

In Figure 22 two instances of Element A: Element A1 and Element A2 produce erroneous outputs (Error A1 and Error A2) because both elements are affected by a fault that results from a same root cause. As Element A1 and Element A2 are used as redundant elements compared by Element B, we are in the presence of a possible dependent failure if Error A1 and Error A2 cannot be differentiated by Element B at the time they are compared.

NOTE It is assumed for simplification that Element B itself is not affected by the disturbance. Taking into account the assumption that Element B is operational it is further assumed that as long as Error A1 and Error A2 present some temporal or spatial dissimilarity, the dependent failure situation can be controlled. Such dissimilarity can be the consequence of differences in the manner the disturbance propagates to both elements (e.g. different propagation delay of a signal glitch that takes different physical routes to reach the boundaries of Element A1 and Element A2) or in differences in the effect (e.g. if the effect is a signal timing violation, it can have different effect on the respective logic of Element A and Element B).

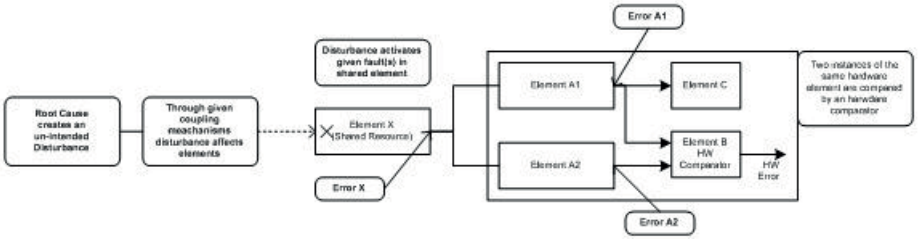

Figure 23 — Dependent failure due to resource sharing

Figure 23 extends Figure 22 where Element A1 and Element A2 produce erroneous outputs caused by an erroneous output of the shared Element X that is affected by a fault that results from a root cause external to the element itself. The erroneous output of Element X propagates to both Element A1 and Element A2. Element X is representative of the dependent failure initiators that fall into the category "Shared Resources".

10.4 Distinction between cascading failures and common cause failures

Except for ISO 26262-9:2011, Clause 6, no distinctions between common cause failures and cascading failures are necessary. Since the exact differentiation between a cascading failure and a common cause failure in a given failure scenario is not always possible, the two failure scenarios are not differentiated any further within this document. Instead they are merged into dependent failures. If the focus of the DFA is to show freedom from interference between two given elements (e.g. Element A and Element B) as requested in ISO 26262-9:2011, Clause 6, the following approach is considered sufficient:

— Identify the failure modes of Element A which can have an impact on Element B.

— Identify if these failure modes lead to a possible violation of the safety goal due to the failure of Element B.

— If necessary define appropriate safety measures to mitigate the risk (e.g. for a DMA specify a safety mechanism that monitors the addresses generated by the DMA).

— If necessary repeat this analysis with switched roles.

10.5 Dependent failure initiators

10.5.1 Dependent failure initiator list

The following classification of DFI is proposed:

— Failure of shared resources;

— Single physical root cause;

— Environmental faults;

— Development faults;

— Manufacturing faults;

— Installation faults;

— Repair faults.

NOTE 1 Other classifications of DFI are possible.

For each class of dependent failures possible measures[2)] are provided. The measures have been split into measures which prevent the dependent failure occurring during operation and into measures which do not prevent the occurrence of the dependent failure but prevent it from violating a safety goal.

NOTE 2 DFI that are caused by software are not included in this DFI list. Correct software development is addressed by ISO 26262-6. Results of the DFA can affect the ASIL allocation of software elements.

NOTE 3 Repair in automotive typically happens on the level of ECUs, not on the level of semiconductor components. Therefore repair faults are usually no DFI for semiconductor parts.

2) The listed measures are examples provided as a non-exhaustive list of possible solutions. Their efficiency depends on several factors including type of circuits and technology, and they could be not effective at the same way for all possible DFI. For that reason, evidences are recommended to demonstrate the claimed efficiency. Some measures by themselves might not be enough to achieve an appropriate risk reduction. In this case an appropriate combination of different measures can be chosen.

Table 47 — Dependent failure initiators due to random hardware faults of shared resources

DFI examples	Measures to prevent dependent failures from violating the safety goal	Measures to prevent the occurrence of dependent failures during operation
Common clock (including PLL, clock trees, clock enable signals, etc...) Common test logic including DFT (Design for Test) signals, scan chains etc..., common debug logic including debug routing network (network that provides access to analogue or digital signals or allows to read digital registers) and trace signals (mechanism to trace one or more signals synchronously (e.g. controlled by triggers or trace clocks and read the result afterwards) Power supply elements including power distribution network, common voltage regulators, common references (e.g. band-gaps, bias generators and related network) Non simultaneous supply switch-on, that can cause effects like latch up or high in-rush current Common reset logic including reset signals Shared modules (e.g. RAM, Flash, ADC, Timers, DMA, Interrupt Controller, Busses, etc...)	Dedicated independent monitoring of shared resources (e.g. clock monitoring, voltage monitoring, ECC for memories, CRC over configuration register content, signalling of test or debug mode) Confirmation that dependent failure does not violate a safety requirement by fault simulation (e.g. to inject faulty scenarios and evaluate the impact) Self-tests at start-up or post-run or during operation of the shared resources Diversification of impact (e.g. clock delay between master and checker core, diverse master and checker core, different critical paths) Indirect detection of failure of shared resource (e.g. cyclic self-test of a function that would fail in case of a failure of the shared resource) Indirect monitoring using special sensors (e.g. delay lines used as common-cause failure sensors)	Fault avoidance measures (e.g. conservative specification), functional redundancies within shared resources (e.g. multiple via/contacts), Fault diagnosis (e.g. ability of identifying and isolating or reconfiguring/replacing failing shared resources, corresponding design rules) Dedicated production tests (e.g. end-of-line tests for SRAM capable to find complex faults) Separate resources to reduce the amount or scope of shared resources Adaptive measures to reduce susceptibility (e.g. voltage/operating frequency decrease)

Table 48 — Dependent failure initiators due to random physical root causes

DFI examples	Measures to prevent dependent failures from violating the safety goal	Measures to prevent the occurrence of dependent failures during operation
Short circuits due to e.g.: local defects, electro migration, via migration, contact migration, oxide break down Latch up Cross talk (substrate current, capacitive coupling) Local heating caused e.g. by defective voltage regulators or output drivers	Confirmation that dependent failure does not violate a safety requirement by fault simulation (e.g. to inject faulty scenarios and evaluate the impact) Diversification of impact (e.g. clock delay between master and checker core, diverse master and checker core, different critical paths) Indirect detection (e.g. cyclic self-test of a function that would fail in case of a failure of the shared resource) or indirect monitoring using special sensors (e.g. delay lines used as common-cause failure sensors)	Dedicated production tests Fault avoidance measures (e.g. physical separation/isolation, corresponding design rules) Physical separation on a single chips

Table 49 — Systematic dependent failure initiators due to environmental conditions

DFI examples	Measures to prevent dependent failures from violating the safety goal	Measures to prevent the occurrence of dependent failures during operation
Temperature Vibration Pressure Humidity / Condensation Corrosion EMI Overvoltage applied from external Mechanical stress Wear Aging	Diversification of impact (e.g. clock delay between master and checker core, diverse master and checker core, different critical paths) Direct monitoring of environmental conditions (e.g. temperature sensor) or indirect monitoring of environmental conditions (e.g. delay lines used as dependent -failure sensors)	Fault avoidance measures (e.g. conservative specification/robust design) Physical separation (e.g. distance of the die from a local heat source external of the die) Adaptive measures to reduce susceptibility (e.g. voltage/operating frequency decrease) Limit the access frequency or limit allowed operation cycles for subparts (e.g. specify the number of write cycles for an EEPROM)

Table 50 — Systematic dependent failure initiators due to development faults

DFI examples	Measures to prevent dependent failures from violating the safety goal	Measures to prevent the occurrence of dependent failures during operation
Requirement faults Specification errors Implementation faults, Incorrect implementation of functionality Lack or insufficiency of design measures to avoid crosstalk Lack or insufficiency of Latch up prevention measures Wrong microcontroller configuration Layout faults, such as incorrect routing e.g. over redundant blocks, insufficient insulation, insufficient distance, insufficient EMI shielding Temperature due to local heating of power consuming parts of the die Temperature gradients causing mismatches within sensitive measurement circuitry	Monitors (e.g. protocol checkers) Fault simulation (e.g. to stimulate faulty scenarios and evaluate the impact)	ISO 26262 compliant design process Diversity (Depending on the DFI, diversity can be intended either as implementation/ functional/architectural diversity or as development diversity)

Table 51 — Systematic dependent failure initiators due to manufacturing faults

DFI examples	Measures to prevent dependent failures from violating the safety goal	Measures to prevent the occurrence of dependent failures during operation
Related to processes procedures and training Faults in control plans and in monitoring of special characteristics Related to software flashing and end-of-line programming (e.g. wrong versions or wrong programming conditions, protocols or timings) Mask misalignment Incorrect End-of-Line trimming or fusing (e.g. Laser trimming, OTP or EEPROM programming of calibration coefficients or customization settings)		Dedicated production tests ISO 26262 compliance (e.g. part 7) Diversity (Depending on the DFI, diversity can be intended either as implementation/functional/architectural diversity or as development diversity)

Table 52 — Systematic dependent failure initiators due to installation faults

DFI examples	Measures to prevent dependent failures from violating the safety goal	Measures to prevent the occurrence of dependent failures during operation
Related to wiring harness routing Related to the inter-changeability of parts Failures of adjacent items or parts or elements. (e.g. wrong configuration of a connected interface delivering data to an input or incorrect load on a driven output) Wrong microcontroller PCB connection Wrong configuration (e.g. of spare memory usage)		Dedicated installation tests ISO 26262 compliance (e.g. part 7) Diversity (Depending on the DFI, diversity can be intended either as implementation/ functional/architectural diversity or as development diversity)

10.5.2 Verification of mitigation measures

This section introduces exemplary methods to evaluate the effectiveness to control or avoid dependent failures. The methods can be based on:

— Analytical approach using known principles;

EXAMPLE 1 Reference[10] and similar provide analytical approaches that can be used as a basis to evaluate the effectiveness of the provided safety mechanisms addressing dependent failures

— Pre-silicon simulation using documented test protocols to show robustness against the identified DFI;

EXAMPLE 2 Test protocols that allow simulation of clock or power supply disturbances, EMI simulations etc. The simulation can be based on different levels of abstraction (based on the fault model to be targeted) and use adequate fault injection techniques to produce the intended disturbance.

— Post-silicon robustness tests (e.g. EMI test, burn In studies, accelerated aging test, electrical stress tests)

— Expert judgment supported by documented rationale;

A combination of measures can be used, e.g. references[32],[29] and similar provide a mix of analytical and fault injection based approaches that can be used as a basis to evaluate the effectiveness of the provided safety mechanisms addressing dependent failures.

NOTE The use of beta factors as in IEC 61508–2:2010[21] for the quantification of coupling effects is not foreseen in ISO 26262-9:2011, 7.4.2.

The level of detail of the evaluation is commensurate with the targeted ASIL, the claimed safety mechanisms and application.

EXAMPLE 3 A higher level of detail is used in case of on-chip safety mechanisms or necessity to provide fail operational functionality

As stated in the example in ISO 26262-9:2011, 7.4.7, diversity is a measure that can be used to prevent, reduce or detect common cause failures. In case diversity is used as a method to control or avoid

dependent failures, a rationale is provided to demonstrate that the level of implemented diversity is commensurate to the targeted DFI.

EXAMPLE 4 Rationale can be provided with a combination of analytical approach and fault injection (e.g. as described in reference[32])

In case isolation or separation is used as a method to control or avoid dependent failures, a rationale is provided to demonstrate that the level of implemented isolation or separation is commensurate to the targeted DFI.

EXAMPLE 5 Rationale can be provided with a simulation to show that the distance between two separated blocks is commensurate to avoid the targeted DFI

EXAMPLE 6 A higher level of detail is used in case of on-chip safety mechanisms or necessity to provide fail operational functionality

10.6 DFA workflow

The purpose of the DFA workflow is to identify the main activities that are judged necessary to understand the operation of the safety mechanism that are implemented to ensure achievement of the safety requirements and verify that they meet the requirements for independence.

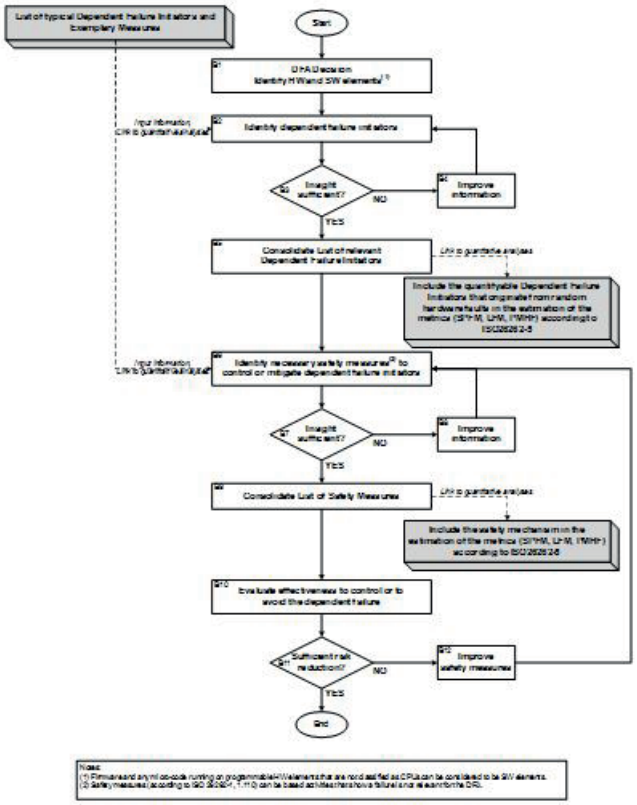

Figure 24 — DFA workflow

10.6.1 DFA decision and identification of HW and SW elements (B1)

A DFA according to ISO 26262-9:2011, Clause 7 for a semiconductor element should be conducted in any case that may require independence or freedom from interference e.g.:

— Diagnostic functions assigned to hardware or software elements.

— Similar or dissimilar redundancy of hardware or software elements.

— Shared resources on the hardware component or part (e.g. clock, reset, supply memory, ADC, I/O, test logic).

— Execution of multiple software tasks on shared hardware.

— Shared software functions (e.g. I/O-routines, interrupt handling, configuration).

— Required physical separations by distance or any kind of barriers (e.g. supply of different voltage domains).

— Independence requirements derived from ASIL decomposition on system level that affect different elements on the IC, where the DFA needs to provide evidence of sufficient independence in the design or that the potential common causes lead to a safe state. (Refer to ISO 26262-9:2011, Clause 5).

The input to this step are the technical safety requirements that are introduced up to here, the derived safety concept and a description of the architecture (including block diagrams, flow charts, fault trees, state diagrams, software partitioning) of the implementation.

The focus of this step is to analyse the architecture and identify all pairs or groups of elements that can be concerned by any of the above listed cases and evaluate if the architectural description is detailed enough to capture the overall design dependencies. The outcome of this step is a list of all pairs or groups of elements that can be affected by dependent failure and associated independence or freedom from interference requirements.

10.6.2 Identification of DFI (B2)

This step is based on the prior architectural analysis and targets a check of the completeness of the derived independence or freedom from interference requirements and break them down wherever different initiators can lead to a dependent failure.

A list of typical DFI as provided in 10.5.1 can be used to prove whether known dependent failures other than the ones that were derived from the architecture can be applied. Further it is crosschecked if dependent failure mechanisms were identified during the quantitative analysis.

The outcome of this step is a consolidation of the list from the previous step.

10.6.3 Sufficiency of insight provided by the available information on the effect of identified DFI (B3 and B4)

In this step it is verified if the available documentation provides sufficient insight to all DFI that where evaluated during previous steps. In case any additional information is required to judge the validity of a DFI for the target architecture, it is added and the identification of the DFI (step 2) can be finished based on the updated descriptions.

NOTE A hierarchical approach is recommended so that the analysis can be done on an appropriate level of detail. For instance a top level view enables to understand what the shared resources are. Then a breakdown view that encapsulates a hardware sub-part and its safety mechanisms can be used to identify dependencies at the design level.

10.6.4 Consolidation of list of relevant DFI (B5)

Based on the provided consolidated information, the list of identified DFA relevant elements, independence requirements and the related fault initiators for the fulfilment of the safety requirements is consolidated (e.g. by review).

From the consolidated list dependent failure mechanisms that are caused by random hardware faults can be incorporated into the quantitative analysis of the required metrics (SPFM, LFM, and PMHF).

10.6.5 Identification of necessary safety measures to control or mitigate DFI (B6)

In order to fulfil the safety requirements as described in the functional safety concept necessary safety measures shall be added to mitigate the effect of the dependent failures that are relevant for the target architecture.

Examples of measures that are usually effective to mitigate DFI are given in the list of typical DFI in 10.5.1. Finally the required safety mechanisms shall be integrated into the documentation of the safety concept and the architecture to implement it.

NOTE 1 For dependent failures that arise from random hardware faults the result of the quantitative analysis can be used to identify the ones that are relevant to achieve the targeted metrics (SPFM, LFM, and PMHF).

NOTE 2 If quantifiable random hardware failure are identified to be relevant as DFI (e.g. a shared oscillator delivering a clock that is too fast for the timing constraints of a digital core; overvoltage delivered to an internal supply due to a fault of a supply voltage regulator) they are taken into account for the quantitative analysis (see ISO 26262-5:2011, 9.4.3.2, NOTE 1). For the case that they are not quantifiable (e.g. the influence of timing effects caused by a fault in a clock tree; thermal coupling effects between an element and its safety mechanism; substrate currents due to a fault in one of the blocks that will be independent) the evaluation and definition of mitigation measures is continued qualitatively (see ISO 26262-9:2011, 7.4.2).

10.6.6 Sufficiency of insight provided by the available information on the defined mitigation measures (B7 and B8)

In this step it is verified if the available documentation provides sufficient insight to analyse the effectiveness of safety measures that were introduced during the previous step. For the case that any additional information is required to judge the mitigation of a DFI for the target architecture including all safety mechanisms, it is added and the definition of dependent failure mitigation measures is finished based on the updated descriptions.

10.6.7 Consolidate list of safety measures (B9)

The list of the defined safety measures for the mitigation of dependent failures is consolidated based on the updated documentation (e.g. by review).

NOTE 1 For dependent failure mechanisms that were incorporated into the quantitative analysis (see B5) the effect of the safety mechanism can also be evaluated quantitatively.

NOTE 2 Additional safety mechanisms which are introduced to mitigate DFI, independently if they were introduced due to quantitative or qualitative evaluation, change the chip area and thus influence the failure rate distribution over all parts of the chip. Thus the quantitative analysis usually is updated.

10.6.8 Evaluation of the effectiveness to control or to avoid the dependent failure (B10)

In order to close the DFA the effectiveness of the introduced safety measures to mitigate or avoid dependent failures shall be verified. The verification methods that can be applied are identical to those that are applied in case of safety mechanisms defined to avoid random or systematic failures according to ISO 26262-5:2011, Clause 10 and Annex D and ISO 26262-6:2011, Clause 10, Clause11, and Annex D. The following techniques can be useful:

— FTA, ETA, FMEA;

— Fault injection simulation;

— Application of specific design rules based on technology qualification tests;

— Overdesign with respect to e.g. device voltage classes or distances;

— Stress testing with respect to temperature profile or overvoltage of supply and inputs;

— EMC and ESD testing.

The verification of safety measures that were integrated into the quantitative analysis can be done in the quantitative analysis as well and sufficient improvements of the resulting metrics can be verified according to ISO 26262-5, Clause 9 and Annex D.

NOTE 1 For the case that an introduced safety measure can be subject of dependent failures as well, their avoidance or mitigation will be evaluated by (re)applying the DFA procedure for the newly introduced dependent failures.

NOTE 2 If there is proven experience with similar measures to mitigate dependent failures, it can be used to judge effectiveness of the measure under analysis, given that the transferability of the result can be argued

10.6.9 Assessment of risk reduction sufficiency and if required improve defined measures (B11 and B12)

To close the DFA an evaluation of the remaining risks of dependant failures is completed. If the mitigation is not regarded to be sufficient, the safety mechanism is improved (B12) and the evaluation of the effectiveness is repeated.

For the case that residual risks can be quantified, they are to be accounted in the quantitative analysis (if not already done in the quantitative analysis path via B5 and B9). For example in case of a function and its safety mechanism which are affected by a dependent failure, the failure mode coverage of the safety mechanism shall be reduced accounting for the unmitigated dependencies.

NOTE If the targeted metrics of quantitative analyses are achieved, risk is understood as sufficiently low from the random hardware fault point of view, even if no safety mechanism is allocated to the hardware element which is affected by the fault that was identified as relevant DFI. Systematic DFI concerning the same element are handled in the DFA on a qualitative base and can lead to the definition of safety measures independent of the quantitative analysis result.

10.7 Examples of dependent failure analysis

10.7.1 Microcontroller example

10.7.1.1 Description

The microcontroller component in Figure 25 is used to illustrate the dependent failure analysis methodology for a digital component.

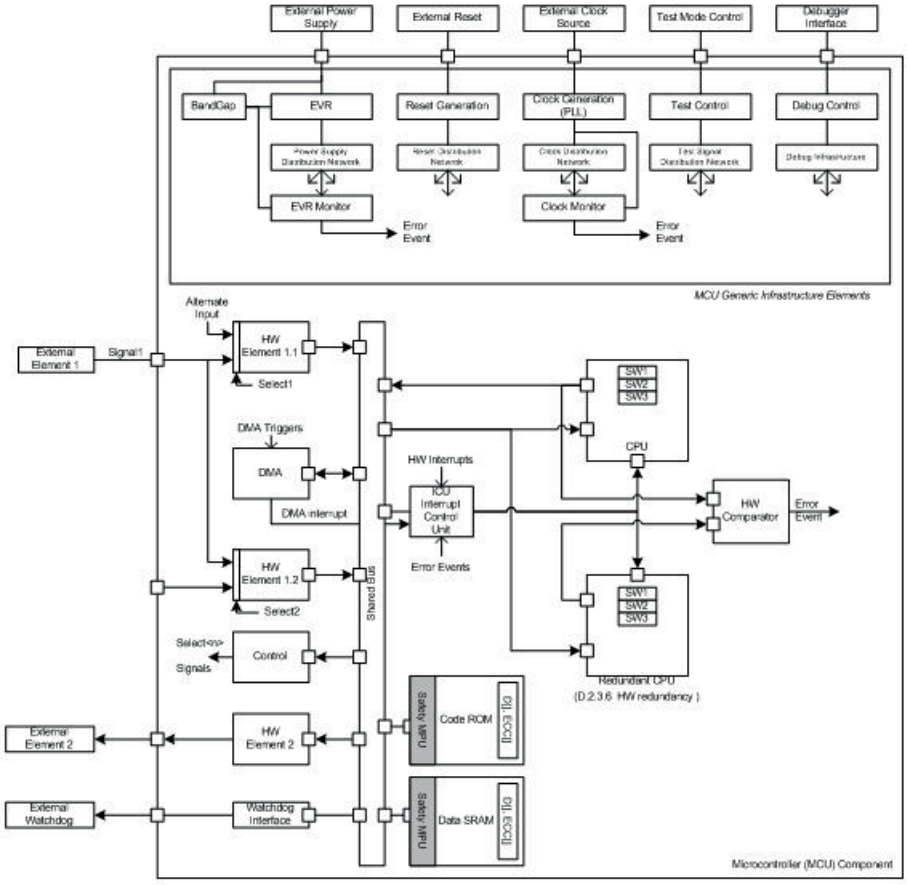

Figure 25 — Microcontroller component example

First an introduction to the hardware and software elements is done to highlight the hardware safety mechanisms that are going to be used for the DFA. It is not in the scope of this example to provide a comprehensive specification of the hardware safety requirements and the safety mechanisms.

— HW Element 1.1: Interface processing element that enables to receive information from HW elements connecting to the Microcontroller (e.g. Signal 1 from External Element 1 in Figure 25).

— HW Element 1.2: Interface processing element identical to HW Element 1.1 from a functional point of view.

— HW Element 2: This element is used to control the External Element 2.

— Control: This element provides the select signals that enable to control the connectivity of HW Element 1.1 and 1.2 with different input interfaces of the microcontroller.

— CPU: Central Processing Unit where all the software elements execute.

— Data SRAM: Memory where software elements store their own private variables. It also contains communication buffers between software and DMA and between software elements themselves.

— Code ROM: Read-only Memory containing the code that is executed by the software elements and possibly constant data used by the software elements.

— Software Elements: In this example three software elements are listed: SW1, SW2 and SW3.

— Watchdog Interface: It enables to communicate with an external watchdog hardware element.

— Shared Resources: The following shared resources are identified:

  — DMA (Direct Memory Access) HW Element: The DMA can be used by all software elements and has read and write access to any addressable resource (Memory, Configuration Register)

  — EVR (Embedded Voltage Regulator): The EVR provides the power supply to all the HW elements inside the microcontroller with the exception of the input/output pads that are powered by the "External Power Supply".

  — Reset Generation and Distribution: Controls the reset state of the microcontroller based on reset commands originating from the external reset source or internal reset actions controlled by hardware or software elements

  — Clock Generation and Distribution: Delivers the intended clocks for all hardware elements based on a PLL using an External Clock Source".

  — Test Logic: Test structures required for the production tests of the microcontroller.

Functional safety concept and requirement concept: Signal S1 is an analogue signal that indicates the state of an actuator. An unintended state shall be recognized and shall lead to the de-activation of the actuator: this is considered to be the safe state. For that purpose the Signal S1 is converted into digital information and then processed by a software element SW1 to identify a possible hazardous state of the actuator. The software element SW2 is responsible to redundantly acquire information from Hardware Element 1.1 and 1.2. The main task of SW2 is to control the DMA to fetch the conversion results from Hardware Element 1.1 and 1.2 and store them in separated data sets in a shared buffer located in Data SRAM. DMA informs SW2 about the completion of transfers by sending an interrupt to the ICU. Upon reception of this event SW2 compares the plausibility of the data sets and in case of mismatch it provides pre-defined error information to SW1. The software element SW3 is responsible for a periodic refresh of the external watchdog. The refresh requires sending a dynamic code with a given sequence. The code to be sent is only provided by software element SW1. If SW3 fails to refresh the watchdog or sends an incorrect code, the external watchdog enters timeout state that leads to the de-activation of the actuator.

This clause provides exemplary safety requirements. The specification of the set of safety requirements is reduced to a minimum set that is suitable for the DFA.

— MCU-REQ-1: Faults during the processing of Signal 1 by HW element 1.1 shall be detected within 20 ms [ASIL X].

— MCU-REQ-1.1: Signal 1 shall be redundantly processed by HW Element 1.2.

— MCU-REQ-1.2: Results of HW Element 1.1 and 1.2 shall be monitored by SW. In the presence of a mismatch SW shall send an error message to the external watchdog through the watchdog interface.

— MCU-REQ-2: Random hardware fault leading to a wrong output of CPU shall be detected within 20 ms [ASIL X].

— MCU-REQ-2.1: CPU shall be monitored by a redundant CPU. Outputs of CPU and Redundant CPU shall be compared every clock cycle by an HW comparator.

— MCU-REQ-2.2: In the presence of a mismatch between CPU and Redundant CPU an error event shall be generated.

10.7.1.2 Dependent failure analysis

The DFA will only focus on the DFI that have the potential to lead to a violation of the safety requirement MCU-REQ-2. The analysis will follow the proposed workflow. To simplify the analysis not all the steps will be considered.

With respect to the requirements MCU-REQ-2, this step focuses on analysing the architecture focusing on steps B1 and B2 of the DFA workflow. The analysis is supported by a qualitative fault tree that identifies the shared resources and the redundant elements.

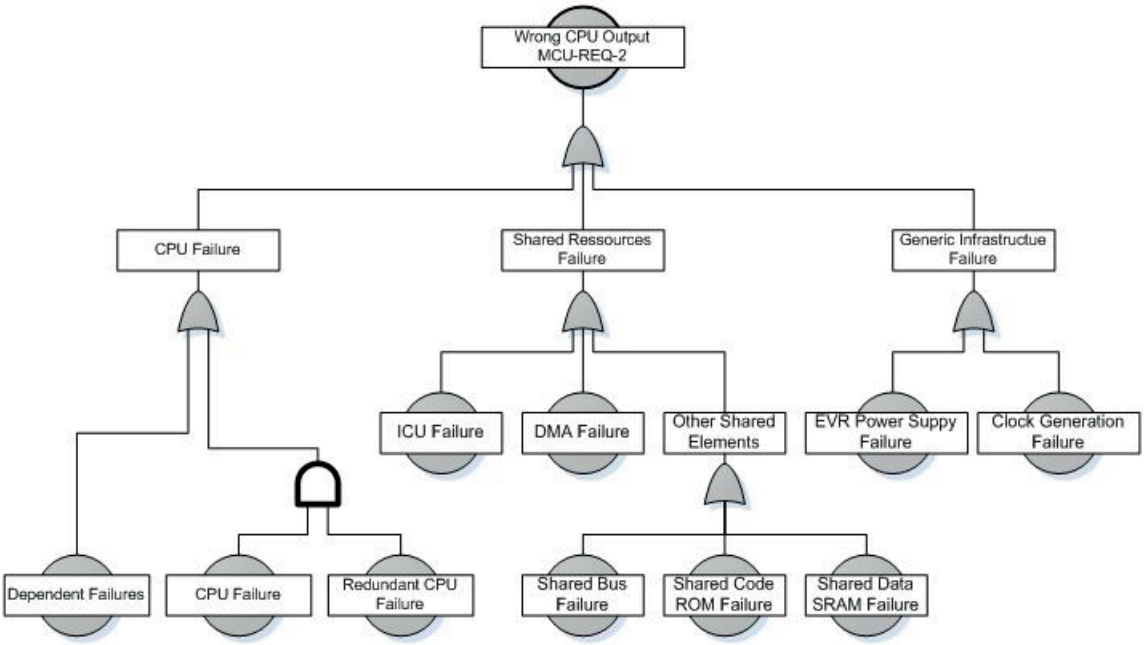

Figure 26 — Shared elements overview

For the shared resources, each failure base event or AND gate needs to be analysed on its own. For the CPU and redundant CPU a base event Dependent Failures has already been introduced because the safety mechanism is already visible on the proposed architectural level. It is recommended to analyse the Generic Infrastructure Elements that have a global effect separately, in order to avoid considering them for each shared element independently. This is possible for the power supply and clock generation because they have own safety mechanisms. However for the Reset Generation, Test Signals and Debug Infrastructure it is necessary to analyse them at a lower level where their influence to the shared elements safety mechanisms can be analysed. For the Generic Infrastructure Elements the analysis will concentrate on the power supply and clock generation.

Table 53 shows an example for a microcontroller DFA:

Table 53 — DFA for microcontroller example

ID	Element	Redundant element	Dependent failure initiators			DFA	
	Short name and description	Short name and description	Systematic faults	Shared resources	Single physical root cause	Measure for fault (A) voidance or (C) ontrol	Verification method
Generic Infrastructure Elements							
PS1	Power Supply	Power Supply Monitor: Measurement of voltage levels within operating conditions		Shared bandgap has the potential to lead to undetected over voltage.		(C) Add a bandgap monitor	Silicon-level robustness test

Table 53 *(continued)*

ID	Element	Redundant element	Dependent failure initiators			DFA	
PLL1	Clock	Clock Monitor Frequency Measurement		Shared Input Frequency has the potential to prevent accurate Frequency measurement.		(C) Add an independent clock source (Oscillator) to measure the PLL frequency '(A) Design dissimilarity: dissimilarity between drift behaviour of PLL and drift behaviour of reference oscillator used by clock monitor thanks to different implementation.	Design Inspection Silicon-level robustness test
PLL2	Clock	Clock Monitor Frequency Measurement		Loss of clock that prevents monitor to report failure condition		(C) Semiconductor monitoring by External Watchdog Function.	
PLL3	Clock	Clock Monitor Frequency Measurement			Needs to be analysed based on a detailed block diagram of the clock generation and clock monitoring where the relevant interfaces, sideband signals and configuration registers are visible.	< other measures >	< other verification methods >
Processing Elements							
CPU1	CPU, Computation	Redundant CPU + Hardware Comparator		Power Supply		Covered by Power Supply Analysis	
CPU2	CPU, Computation	Redundant CPU + Hardware Comparator		Clock: incorrect frequency		Covered by PLL Analysis	
CPU3	CPU, Computation	Redundant CPU + Hardware Comparator		Clock: clock glitch			
CPU4	CPU, Computation	Redundant CPU + Hardware Comparator		Shared Bus			

Table 53 *(continued)*

ID	Element	Redundant element	Dependent failure initiators			DFA	
CPU5	CPU, Computation	Redundant CPU + Hardware Comparator		Data SRAM		Safety Mechanisms for Data SRAM (e.g. ECC) are covered by Safety Analysis. ECC is evaluated by redundant CPU enabling to control this dependent failures related to interface to Data SRAM.	
CPU6	CPU, Computation	Redundant CPU + Hardware Comparator		Code SRAM			
CPU7	CPU, Computation	Redundant CPU + Hardware Comparator		ICU			
CPU8	CPU, Computation	Redundant CPU + Hardware Comparator			Short-circuit between signals belonging to CPU and signals belonging to Redundant CPU	(A) Physical separation according to technology design rules	Analysis of design rules Physical Layout inspection
CPU9	CPU, Computation	Redundant CPU + Hardware Comparator			Latch-up affecting logic belonging to CPU and logic belonging to Redundant CPU	(A) Physical separation according to technology design rules for isolation of standard cells against latch-up (A) Physical separation related to Soft-Error Induced Latch-up	Analysis of design rules Physical Layout inspection

After the architectural enhancements resulting from the DFA the microcontroller component block diagram is updated as showing the

— new Bandgap Monitor element to mitigate the dependent failures related to the Bandgap drift failure mode

— the new Oscillator element to mitigate the dependent failures related to the clock drift failure mode

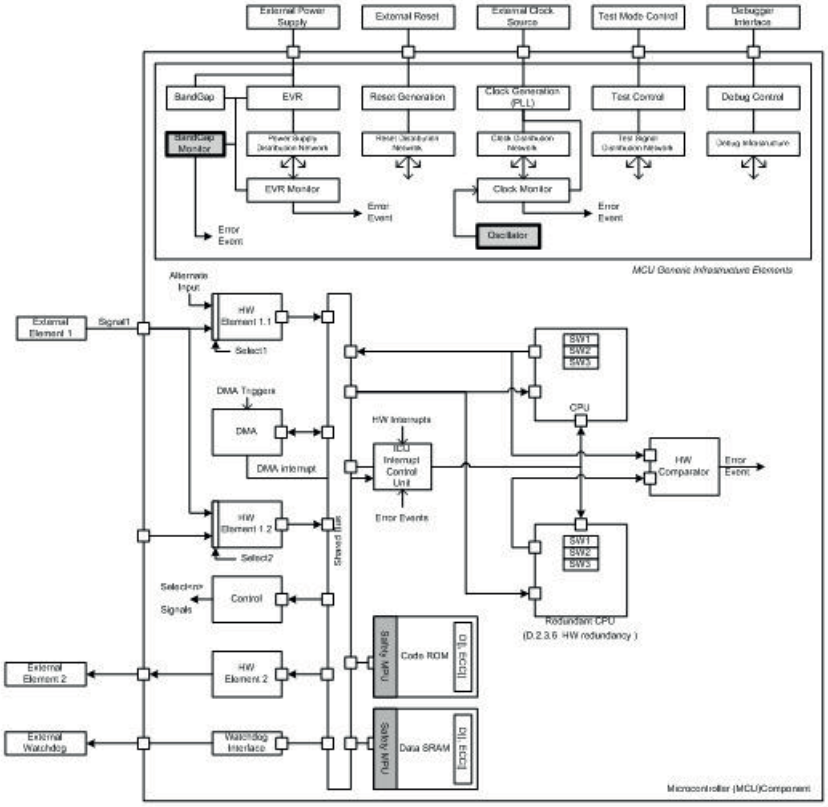

Figure 27 — Enhanced microcontroller component

10.7.2 Analog example

10.7.2.1 Description

The analogue example is intended to provide guidance on the application of a DFA to analogue components, part or subparts. The detailed failure modes, relevant DFI, safety requirements and choice of considered safety and mitigation measures are typical examples, but they are not to be considered as exhaustive and can change depending on the details of the application, system architecture, circuit design and IC-technology.

The DFA of an analogue part is explained in the following chapters based on an assumed architecture of a switched output stage. The architecture of this output stage is sketched in Figure 28. It uses high voltage N-DMOS switch transistors to activate the current path through a load which might for example be part of an actuator in a safety application. In order to avoid that faults of a switch transistor or its gate driver can activate the actuator inadvertently, the switches are redundantly placed in the high side and low side current paths to the load. The high side and low side drivers are supplied by a regulated voltage V_{dd} which is significantly lower than the external supply V_{bat} coming from the board net connected to the 12V battery of the car. The output of the supply voltage regulator is already monitored by a voltage monitor which is used for non-safety purposed like the provision of a power on reset. The gate voltage that is needed to turn on the high side N-DMOS switch transistor is delivered by a charge pump in order to make the driver insensitive to EMC on the board net.

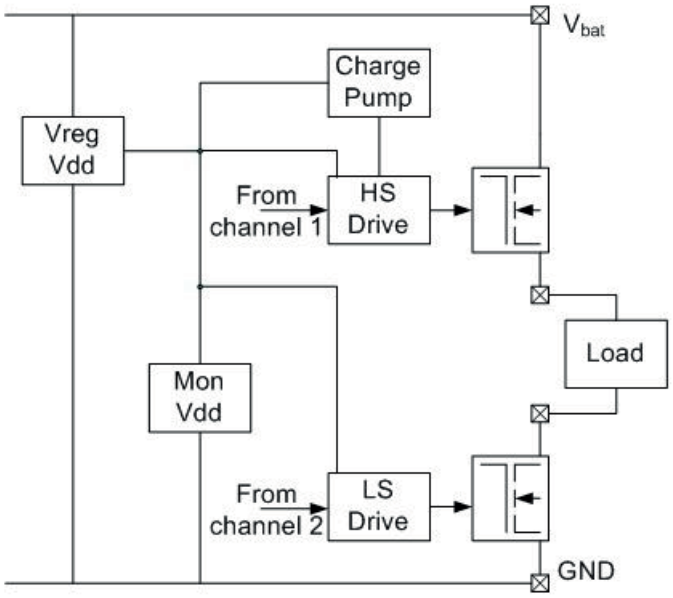

Figure 28 — Analogue output driver example

In order to be able to identify dependent failure mechanisms, the following safety requirement is assumed:

In the inactive state, the load connected between the high side switch transistor output and low side switch transistor output shall not be supplied with a current of more than 1mA for longer than 1ms.

NOTE The current of 1mA is assumed to be much lower than the current that is drawn by the load in the case that the switches are turned on (e.g. 1A).

10.7.2.2 Dependent failure by shared supply voltage regulator

The primary fault that leads to the exemplary dependent failure is illustrated in Figure 29. The supply voltage regulator that supplies the internal driver circuitry for the control of the switch transistor gate voltages fails in a way that the pass device (pass device is the transistor that is in the supply current path) is permanently turned on. The fault mechanism could be a defect of the pass transistor itself or a fault of the control loop that causes instability like e.g. loss of a compensation capacitor. The consequence is a rise of the internal supply level *V*dd to the external supply level V_{bat}.

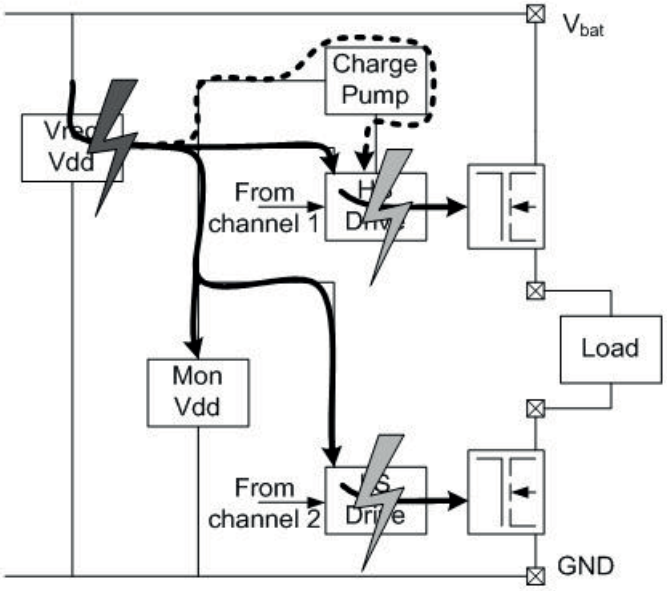

Figure 29 — Dependent failure by shared supply voltage regulator

The fault is assumed to violate the safety requirement in case of its appearance, since the complex driver circuit that we assume for this example cannot be realized in a way that allows operating it shorted to the external supply. Thus severe damage of the driver shall be assumed and the driver output cannot be assumed to keep the gate voltages of the switch transistors on a level that keeps the switch transistors in a high impedance state. Thus the dependent failure that is caused by the "overvoltage" that is applied to the supply of the driver stages is assumed to have worst case consequences for the driver stages. Consequently it propagates to the top level failure in the fault tree shown in Figure 30. In quantitative safety analysis the SPFM of the "overvoltage" failure mode of the supply voltage regulator (not necessarily all failure modes of the supply voltage regulator e.g. under voltage) would be added directly to the SPFM for violating the defined safety goal, as shown by the grey under laid base event for overvoltage from the V_{dd} supply voltage regulator connected to the top level "OR" gate in the FTA.

NOTE There are other dependent failures that could appear as a consequence of overvoltage delivered by the supply voltage regulator. The first one is a fault induced in the charge pump, which is shown as a dotted line in the block diagram. In the worst case this fault can have the same effect than a damage of the high side driver due to overvoltage at its Vdd supply input and is therefore already included in the way the Vdd supply overvoltage fault was introduced in the FTA. Another dependent failure that could be induced by the overvoltage is the damage of the voltage monitor which can cause that the overvoltage stays undetected; this will be handled later on in the discussion of the measures to mitigate the dependent failures of the gate drivers.

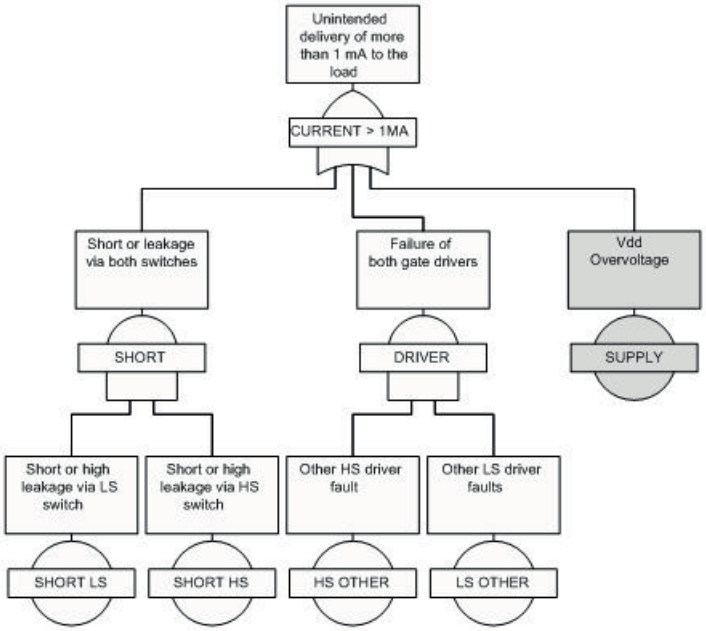

Figure 30 — FTA including shared supply

The following freedom of interference requirement could be derived in order to ensure the achievement of the safety requirement for the case that the described fault in the supply voltage regulator appears.

Freedom from interference requirement could be stated as: "A failure in the supply voltage regulator block shall not cause an activation of either the high side or the low side switch transistor in a way that the corresponding output could deliver a current of more than 1mA to the load for longer than 1ms."

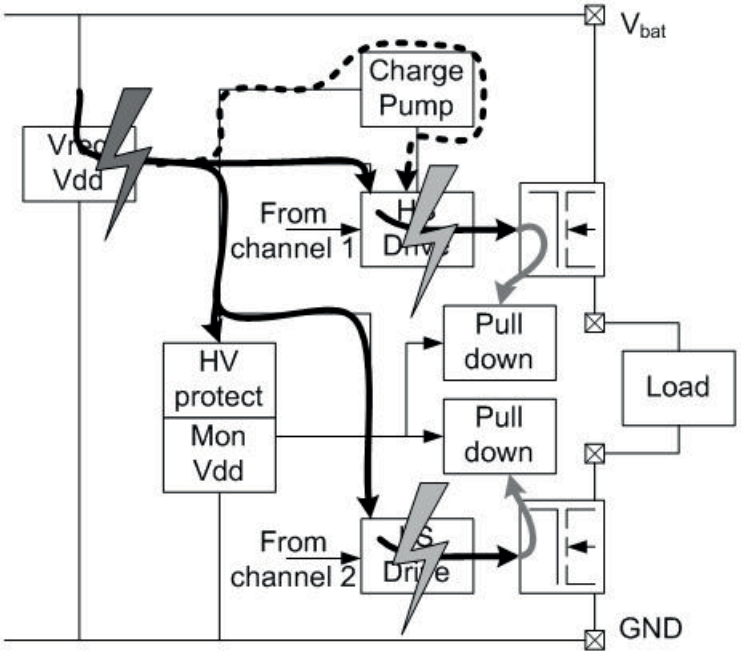

Figure 31 — Shared supply fault mitigation

In order to achieve the demanded freedom from interference, safety measures should be defined in order to avoid a violation of the safety requirement in the case of a connection between the internal supply of the driver stages and the external supply voltage V_{bat}.

Example of taken measures as shown in Figure 31:

— Introduce subparts to pull down the switch transistor gate source voltages below their threshold voltages. The pull down blocks are activated by the supply monitoring block.

— Limit of the current that can pass the connection between the driver output and the switch transistor gate to ensure that the pull down is able to keep the gate source voltage sufficiently low for the case of a short to the supply at the gate driver output.

As a consequence of the introduction of the above mentioned safety mechanisms, the architecture of the system is changed and a rise of the internal supply to the level of the board net is no longer causing a violation of the safety requirement by the initial dependent failure as long as the pull down subparts are activated. If there is no other cascading effect which could impact the function of this safety mechanism the mitigation of the dependent failure would be sufficient.

The adaptation of the fault tree according to the defined mitigation measures that result from the DFA is shown in Figure 32.

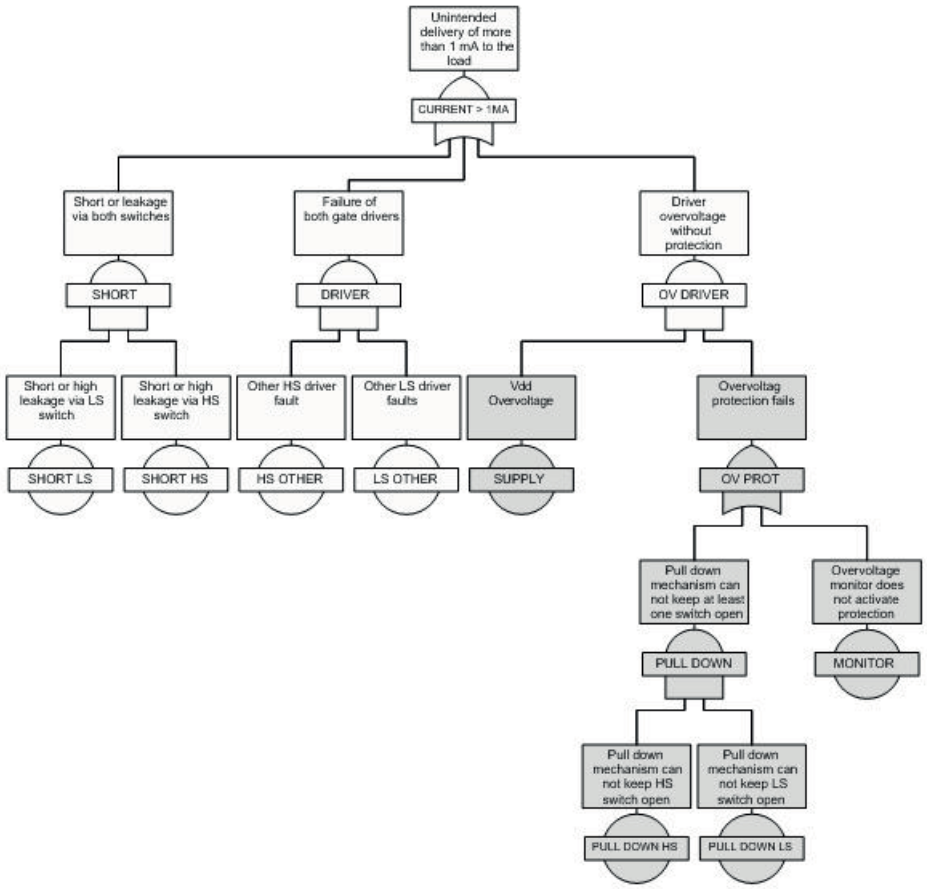

Figure 32 — FTA including shared supply fault mitigation

For the case that the change of the architecture introduces other additional dependent failure mechanisms that could impact the effectiveness of the new safety mechanism (a) and (b) that were introduced to mitigate the initial dependent fault, an additional freedom from interference requirement should be derived. For this case the new freedom from interference requirement could be formulated as follows:

"A failure of the supply voltage regulator that shorts the internal supply V_{dd} to the external supply voltage V_{bat} shall not cause a failure in the voltage monitor or a failure of the pull down blocks, which disables the pull down current paths in a way that the threshold of the switch transistors can be exceeded longer than 1ms."

For the achievement of this new freedom from interference requirement additional safety measures are installed for the switch transistors. These pull down blocks should not be affected by the initial fault (short of the internal supply V_{dd} to the external board net supply V_{bat}) in a way that prevents them from keeping the gates of the output switch transistor pulled down.

Example of taken measures:

— Introduction of a high voltage protection block for the supply monitor

— Design of the gate pull down should be dimensioned for operation with the external supply voltage.

For this example it is assumed that the IC technology allows to implement these measures in a way that provides sufficient safety margin. This assumption is justifiable in a qualitative evaluation, since the supply monitor and the pull down blocks are small and can be realized in a way (e.g. increased channel length, cascaded HV transistors, serial resistors) that allows increased safety margin compared to the supply voltage regulator (higher absolute maximum rating for supply voltage).

Of course the safety requirements, fault mechanisms and suggested mitigation method are just exemplary and based on assumptions of the following boundary conditions:

— a circuit architecture

— application requirements

— capabilities of an IC technology which will be used to fabricate the circuit

The example is used here in order to explain how to perform a DFA of an analogue part and not as reference for the mitigation of dependent failures caused by overvoltage faults of the supply voltage regulator in real switched output stages. Other methods or variants to mitigate the same fault can be exploited instead and should be selected based on the final knowledge of the real boundary conditions (e.g. technology options, external safety mechanisms).

Finally the new subparts that have been introduced to mitigate the dependent failure that was caused by the supply overvoltage should be included in the latent fault analysis. If required they should be tested in repeated time intervals (e.g. at each system start-up) to avoid that they are not functional when the overvoltage fault case appears.

10.7.2.3 Dependent failure by coupling mechanism

The primary fault that leads to the second exemplary dependent failure is illustrated in Figure 33. It is a random hardware fault that appears in the high side driver. It leads to a failure of the high side path, which results in a conductance of the high side switch transistor. It further activates a coupling effect that can initiate a dependent failure in the low side path.

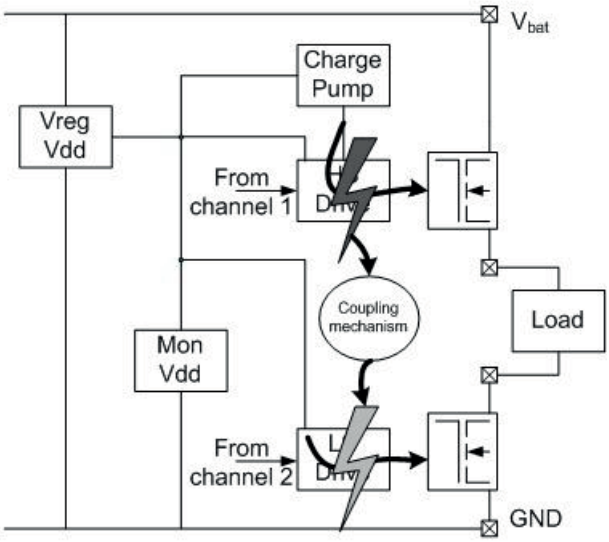

Figure 33 — Dependent failure by coupling mechanism

An independence requirement could be stated as: "A failure of the high side path shall not induce a failure in the low side path that leads to an activation of the low side switch transistor in a way that it can deliver more than 1mA."

As a result of the evaluation of the DFI list we identified the following relevant initiators (see Table 54) and their corresponding coupling mechanisms that require a definition of special mitigation measures.

NOTE 1 This is an example and does of course not imply that these 3 DFI are the only relevant for gate drivers.

Table 54 — Example of identified relevant coupling mechanisms

Reference number	DFI	Coupling mechanism
1	Local hot spot in one of the gate driver circuits (e.g. caused by a defect of an device inside the gate driver block that heats up due to increased power consumption of the defective device).	Heat propagation via the substrate causes an exceed of the maximum rating of the temperature range of the other gate driver.
2	Short circuit in one of the gate drivers leading to a current consumption above the specification of the supply voltage regulator.	Break down of the supply of the other gate driver causes an undefined state (neither within the operating range nor in the range that leads to power on reset).
3	Injection of current into the substrate within one of the gate drivers e.g. caused by defects of substrate pn junctions or by activation of parasitic bipolar transistor of power devices.	Latch up induced including circuit elements of the other gate driver due to increasing voltage drop along the path of the substrate current to GND.

For all dependent failures listed in Table 54 the fault tree in Figure 34 is used. It shows that besides independent random faults in every channel, a coupling between the channels can lead to a fault in the second channel that is not directly affected by the initial fault. In case of temperature increases (reference number 1 in Table 54) or break down of the supply (reference number 2 in Table 54) the dependent failure can be avoided by implementation of a safety mechanism that detects the coupling effect and brings the system into a safe state. In case of the substrate current injection (reference number 3 in Table 54) mitigation could be achieved by technology and/or layout measures that break the coupling mechanism.

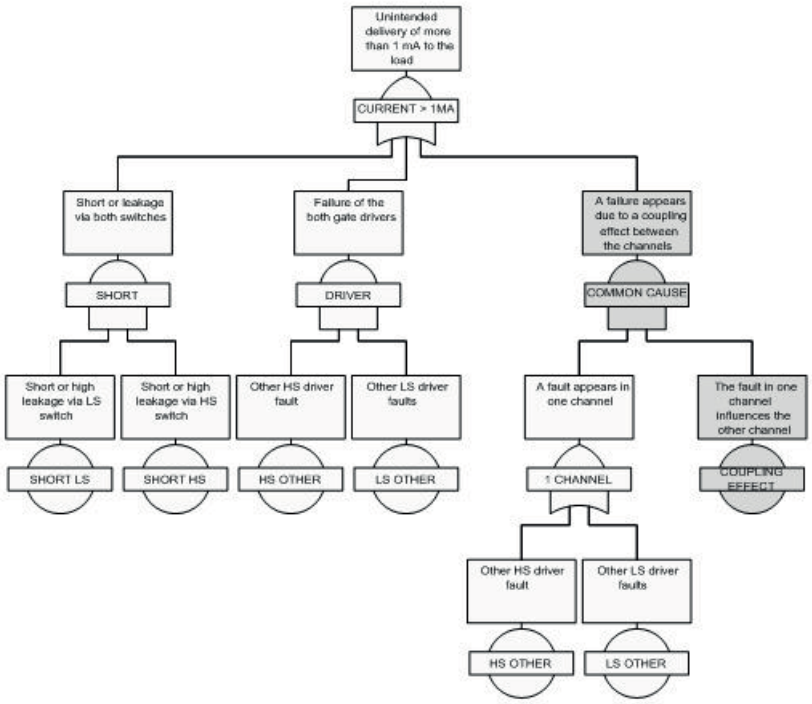

Figure 34 — Fault tree including coupling effect

In order to achieve a mitigation of the identified dependent failures we define additional safety mechanisms in Table 55.

NOTE 2 The mitigation of the dependent failure can require one or a combination of the mitigation measures, a final prove of the evidence of the chosen measures will be provided with respect to the real design, layout, technology, package and application.

Table 55 — Examples for the mitigation of coupling effects

Reference number	Dependent failure mitigation
1	Temperature measurement in the proximity of the gate drivers (the acceptable distance depends on the thermal resistance of the heat sink path and can be found by thermal simulation, sensor elements may be resistors or bipolar transistors) and shut down of the gate driver supply in case of over temperature Current limitation in the supply voltage regulator to limit the power that is available to heat up the chip and brings it into a defined under voltage reset state A thermal segregation (e.g. sufficient distance in combination with a backside heat sink via an exposed die pad) of the independent paths (high side and low side path, each consisting of a switch transistor and its associated gate driver) that is sufficient to prevent the overheating of the fault free path (the one that is not affected by the initial fault). Dimension of the required segregations can be evaluated e.g. based on thermal simulations)
2	Current measurement of the block supplies and shut down of the gate driver supply in case of overcurrent Voltage monitor with under voltage reset that avoids undefined states by setting the reset threshold inside the safe operation range of the circuit Passive pull down of the gates e.g. with resistors to keep switch transistors in off state if the supply is low
3	Physical separation (e.g. spacing, guard rings, separate wells, trenches, buried layer, sinkers – depends on the IC technology) with the target to interrupt the latch up mechanism between the parts that should be independent

Bibliography

[1] ISO 26262-3:2011, *Road vehicles — Functional safety — Part 3: Concept phase*

[2] ISO 26262-4:2011, *Road vehicles — Functional safety — Part 4: Product development at the system level*

[3] ISO 26262-5:2011, *Road vehicles — Functional safety — Part 5: Product development at the hardware level*

[4] ISO 26262-6:2011, *Road vehicles — Functional safety — Part 6: Product development at the software level*

[5] ISO 26262-7:2011, *Road vehicles — Functional safety — Part 7: Production and operation*

[6] ISO 26262-8:2011, *Road vehicles — Functional safety — Part 8: Supporting processes*

[7] ISO 26262-10:2012, *Road vehicles — Functional safety — Part 10: Guideline on ISO 26262*

[8] Askari S., & Nourani M. Design methodology for mitigating transient errors in analogue and mixed-signal circuits. Circuits, Devices & Systems, IET. 2012 Nov., **6** (6) pp. 447–456

[9] Baumann R.C. Radiation-Induced Soft Errors in Advanced Semiconductor Technologies. IEEE Trans. Device Mater. Reliab. 2005 Sep., **5** (3) p. •••

[10] Baruah S.K., & Goossens J. "Rate-monotonic scheduling on uniform multiprocessors", Proceedings of the 23rd International Conference on Distributed Computing Systems 2003, pp.360-366

[11] Börcsök J., Schaefer S., Ugljesa E. "Estimation and Evaluation of Common Cause Failures", Second International Conference on Systems, 2007, ICONS '07, pg.41

[12] Bressoud T.C., & Schneider F.B. "Hypervisor-based fault tolerance", Proceedings of the fifteenth ACM symposium on Operating systems principles, 1995, pp.1–11

[13] Chattopadhyay S., Kee C.L., Roychoudhury A., Kelter T., Marwedel P., Falk H. "A Unified WCET Analysis Framework for Multi-core Platforms", 2012 IEEE 18th Real-Time and Embedded Technology and Applications Symposium, 2012, pp.99-108

[14] Clegg J.R. "Arguing the safety of FPGAs within safety critical systems," Incorporating the SaRS Annual Conference, 4th IET International Conference on Systems Safety, 2009, pp.1-6

[15] Conmy P.M., Pygott C., Bate I. "VHDL guidance for safe and certifiable FPGA design," 5th IET International Conference on System Safety, 2010, pp.1-6

[16] FIDES guide 2009 Edition A (September 2010), "Reliability Methodology for Electronic Systems"

[17] Fleming P.R., Olson B.D., Holman W.T., Bhuva B.L., Massengill L.W. Design Technique for Mitigation of Soft Errors in Differential Switched-Capacitor Circuits. IEEE Trans. Circuits Syst., II Express Briefs. 2008 Sep., **55** (9) pp. 838–842

[18] Franklin M. "Incorporating Fault Tolerance in Superscalar Processors", Proceedings of International Conference on High Performance Computing, Dec. 1996

[19] Hayek A., & Borcsok J. "SRAM-based FPGA design techniques for safety related systems conforming to IEC 61508 a survey and analysis", 2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), 2012, pp.319-324

[20] Heiser G. "The role of virtualization in embedded systems", Proceedings of the 1st workshop on Isolation and integration in embedded systems, IIES '08, 2008, pp.11-16

[21] IEC 61508-2:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems*

[22] IEC 61709:2008, *Electrical components – Reliability - Reference conditions for failure rates and stress models for conversion*

[23] IEC/TR 62380:2004, *Reliability data handbook – Universal model for reliability prediction of electronics components, PCBs, and equipment*

[24] JEDEC - JEP122G (October 2011), *Failure Mechanisms and Models for Semiconductor Devices*

[25] JEDEC - JESD 89A (October 2006), *Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices*

[26] Keckler S.W., Olukotun K., Hofstee H.P. Multicore Processors and Systems. Springer, 2009

[27] Kervarreca G. A universal field failure based reliability prediction model for SMD Integrated Circuits. Elsevier, 2000

[28] Lazzari C. "Phase-Locked Loop Automatic Layout Generation and Transient Fault Injection Analysis: A Case Study" 12th IEEE International On-Line Testing workshop, Como, Italy, July 10-12, 2006, pp. 117-127

[29] Mariani R. "Practical experiences of fault insertion in microcontrollers for automotive applications", 15th IEEE European Test Symposium, ETS2010, May 2010

[30] Mariani R. "Soft errors on digital components", in *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation, Frontiers in Electronic Testing*, Vol. 23, Kluwer Academic Publisher, 2003, pp. 49-60

[31] MIL-HDBK-217, *Military Handbook – Reliability Prediction of Electronic Equipment*

[32] Mitra S., Saxena N.R., McCluskey E.J. Common-mode failures in redundant VLSI systems: a survey. IEEE Trans. Reliab. 2000 Sep., **49** (3) pp. 285–295

[33] Mukherjee S.S. "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor in microarchitecture", MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium, Dec. 2003, pp. 29-40

[34] Niimi Y. "Virtualization Technology and Using Virtual CPU in the Context of ISO26262: The E-Gas Case Study", SAE Technical Paper, April 2013

[35] Paolieri M., & Mariani R. "Towards functional-safe timing-dependable real-time architectures", IEEE 17th International On-Line Testing Symposium (IOLTS), 2011, pp.31-36

[36] Siemens SN 29500, "Failure Rates of Components"

[37] Singh M. "Transient Fault Sensitivity Analysis of Analog-to-Digital Converters (ADCs)", Proceedings of the IEEE Workshop on VLSI (WVLSI '01), 2001

[38] White M., & Bernstein J.B. Microelectronics Reliability: Physics-of-Failure Based Modeling and Lifetime Evaluation. JPL Publ. 2008

ICS 43.040.10
Price based on 123 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Copyright in BSI publications

All the content in BSI publications, including British Standards, is the property of and copyrighted by BSI or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use.

Save for the provisions below, you may not transfer, share or disseminate any portion of the standard to any other person. You may not adapt, distribute, commercially exploit, or publicly display the standard or any portion thereof in any manner whatsoever without BSI's prior written consent.

Storing and using standards

Standards purchased in soft copy format:

- A British Standard purchased in soft copy format is licensed to a sole named user for personal or internal company use only.
- The standard may be stored on more than 1 device provided that it is accessible by the sole named user only and that only 1 copy is accessed at any one time.
- A single paper copy may be printed for personal or internal company use only.

Standards purchased in hard copy format:

- A British Standard purchased in hard copy format is for personal or internal company use only.
- It may not be further reproduced – in any format – to create an additional copy. This includes scanning of the document.

If you need more than 1 copy of the document, or if you wish to share the document on an internal network, you can save money by choosing a subscription product (see 'Subscriptions').

Reproducing extracts

For permission to reproduce content from BSI publications contact the BSI Copyright & Licensing team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email subscriptions@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Useful Contacts

Customer Services
Tel: +44 345 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 345 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.