Agenda

- Introduction
  - Functional Safety Requirements
  - SafeAssure Program
- Role of the Semiconductor Supplier
  - System Challenges
  - Freescale System Solutions
- Safety Concepts of Freescale’s Auto MCUs
  - Integrated Safety Architecture Example
- Safety Software
- Safety Support
  - Dynamic FMEDA
  - System level (beyond MCU)
- Summary
The problem
## The World of Functional Safety Standards

<table>
<thead>
<tr>
<th>Year</th>
<th>Aeronautic</th>
<th>Rail Transport</th>
<th>Generic Standard</th>
<th>Industrial Automation</th>
<th>Automotive</th>
<th>Medical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>DO 178</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DO 178A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1985</td>
<td></td>
<td>CEN 11801</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td></td>
<td></td>
<td>EN 50155</td>
<td>IEC 61508</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td></td>
<td></td>
<td>EN 5012X</td>
<td>IEC 61508</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td>EN 50159</td>
<td>IEC 61508</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td></td>
<td></td>
<td></td>
<td>IEC 61508</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select Freescale products are being defined and designed from the ground up to comply with IEC 61508 ed2.0 (2010-04) and ISO 26262 (2011-11-15).

ISO 26262-1:9 published 15th Nov 2011
ISO/FDIS 26262-10 published 9th Mar 2012
ISO 26262 is changing the Automotive Market

• The market trends have one thing in common: If the underlying systems fail, humans can be put at risk

• Functional Safety means “absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems”

• ISO 26262 is the International Standard for Functional Safety. It is applicable to safety-related automotive systems that include one or more E/E systems and that are installed in series production passenger cars with a max gross weight up to 3.5t”

• ISO 26262 addresses
  - architectural & functional aspects
  - procedural aspects (incl. safety lifecycle)
  - to avoid systematic faults and to control random faults

• Safety management is needed from the start of the product development

• Functional Safety will become a standard requirement in future RFQ’s, across most applications
The ISO 26262 standard

- provides an automotive safety lifecycle which outlines handling of safety system development and operation from project initiation to system decommission
- provides an automotive specific risk-based approach for determining risk classes based on severity, exposure (probability) and controllability of the hazard
- uses four Automotive Safety Integrity Levels (ASIL) for specifying the item’s safety requirements
  - ASIL A: the lowest ASIL level
  - ASIL B: at least 90% SPF and at least 60% latent fault (LF, a fault that isn’t detected but doesn’t lead directly to violation of a safety goal) being detected
  - ASIL C: at least 99% SPF and 90% LF detected
  - ASIL D: the highest ASIL level, at least 99.9% SPF and 99% LF detected
- provides requirements for validation and confirmation requirements to ensure the required safety level is achieved
## Determining the ASIL of an Item

<table>
<thead>
<tr>
<th>Class of severity</th>
<th>Class of probability of exposure regarding operational situations</th>
<th>Classes of controllability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Classes of controllability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C1 (simple)</td>
</tr>
<tr>
<td>S1 (Light and moderate injuries)</td>
<td>E1 (very low)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2 (low)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3 (medium)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4 (high)</td>
<td>QM</td>
</tr>
<tr>
<td>S2 (Severe and life threatening injuries [survival probable])</td>
<td>E1 (very low)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2 (low)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3 (medium)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4 (high)</td>
<td>A</td>
</tr>
<tr>
<td>S3 (Life threatening injuries, fatal injuries)</td>
<td>E1 (very low)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2 (low)</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3 (medium)</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>E4 (high)</td>
<td>B</td>
</tr>
</tbody>
</table>

(QM: “quality managed” → no requirements from standard applied explicitly)
Automotive System Trends for ASIL Levels

Source: Freescale: Expectations based on global customer feedback
**SafeAssure Approach: The Four Key Elements**

### ISO 26262
- **Automotive Industry standard,** adaptation of IEC 61508 for electrical/electronic systems within road vehicles
- **Integrity levels** ASIL A, ASIL B, ASIL C, ASIL D
- **Pub date:** 2011

### IEC 61508
- **Generic Industry standard,** applicable to electrical / electronic / programmable electronic safety-related systems.
- **Integrity levels** SIL 1, SIL 2, SIL 3, SIL 4
- **Pub date:** More than 10 years ago

### People
- Regional functional safety experts

### Documentation
- Safety Application Notes / Safety Manual / FMEDA

### Automotive Software
- AUTOSAR OS & MCAL
- Core Self Test
- Device Self Test; Complex Drivers

### Software Partnerships
- Partnering with leading third-party software providers for automotive and industrial

### Safety Analysis
- Selected products defined & designed from the ground up with safety analysis being done at each step of the process
- Assessments / Audits
- Safety Confirmation Measures

### Continuous Improvement
- Process evaluation, assessments / audits and gap analysis exist to ensure processes are continually optimized

---

**Freescale Quality Foundation**

**Microcontrollers**
- Lockstep Cores, ECC on Memories
- Redundant Functions, Internal Monitors, Built In Self Test, Fault Collection & Control

**Analog and Power Management**
- Voltage Monitors, External Error Monitor, Advanced Watchdog, Built In Self Test

**Sensors**
- Timing Checker, Digital Scan of Signal Chains, DS13 or PS15 Safety Data links

**Organization**
- Safety is an integral part of the Freescale world wide organization

**Project Management**
- Configuration & Change Management, Quality Management, Requirements Management, Architecture & Design, Verification & Validation

**Quality Management**
- ISO TS 16949 Certified Quality Management System
- Hardware - Zero Defects
- Software – SPICE Level 3

**Functional Safety Standards**
- **Automotive** ISO 26262
- **Industrial** IEC 61508

---

**Freescale Quality Foundation**

---

**Functional Safety Standards**

**Safety Support**

**Safety Hardware**

**Safety Software**

---

**Freescale Quality Foundation**
Role of the Semiconductor Supplier
Safety Architecture Challenge

- ISO 26262 safety lifecycle defined as top down approach
  - Next level requirements result from previous level
  - In practice also “push-back” due to availability of products with desired functionality and safety measures
- Safety architecture needs to be defined such that it is safe and can be realized in an efficient way
- Possible development options:
  - Commission custom ASICs with application specific safety measures
  - Use off-the shelf components with an integrated safety architecture
    - Many new components emerging in light of ISO 26262 adoption
  - Define major elements of safety architecture at system level
    - Use “standard” off-the shelf components → discrete (component) safety architecture
    - Traditional way of designing a functional safety system
No “best” Safety Architecture exists!

Every embedded application has its very specifics!

Microcontroller are successful due to the general purpose nature (can be adapted to the specifics of an application)

Therefore Freescale provides products with different Multi-Core safety architectures:

- Monolithic integration (safety system on chip)
- Multiple device system level integration (multiple chip ECU)
- Distributed system integration (multiple ECU system)
- ...

Customer may select their most suitable architecture!
Discrete and Integrated HW Safety Architecture

• Discrete HW Safety Architecture
  - Redundancy resolved at system level by means of redundant components
  - Example: Traditional airbag system consisting of MCU and Safing ASIC
  - System designer performs dependent failure analysis

• Integrated HW Safety Architecture
  - Redundancy resolved at system level and at component level with redundant modules within a component
  - Example: Electric power steering with dual-core lock-step uC
  - Component designer performs dependent failure analysis for redundancy at module level
## From Integrated to Discrete Safety Architecture

<table>
<thead>
<tr>
<th>Integrated Safety Architecture</th>
<th>Discrete Safety Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>At system level and at component level</td>
<td>Resolved at system level</td>
</tr>
<tr>
<td>One or more devices contain dedicated safety measures based on an underlying safety concept</td>
<td>Device Level Safety Measures</td>
</tr>
<tr>
<td>No underlying safety concept at device level, typically, however, measures exploitable as safety measures available</td>
<td>None</td>
</tr>
<tr>
<td>Available for integrated safety measures</td>
<td>Device Level Safety Manual</td>
</tr>
<tr>
<td>FMEDA, FTA for dedicated safety measures</td>
<td>Device Level Safety Analysis</td>
</tr>
<tr>
<td>None, typically general supporting information sufficient</td>
<td>None</td>
</tr>
<tr>
<td>Safety case, with complete device level argument for ISO 26262 compliance at device level</td>
<td>Device Level Safety Argument</td>
</tr>
<tr>
<td>Qualification + optional evaluation of measures in development process against systematic faults</td>
<td></td>
</tr>
</tbody>
</table>
## Quantitative ASIL Requirements for HW

<table>
<thead>
<tr>
<th>ASIL A</th>
<th>ASIL B</th>
<th>ASIL C</th>
<th>ASIL D</th>
</tr>
</thead>
</table>
| Discrete HW Safety Architecture | Feasible  
  • Discrete safety architecture | Feasible  
  • Discrete safety architecture using uC & separate watchdog or uC  
  • Functional and temporal alignment between uC & 2\textsuperscript{nd} channel often challenging  
  • Fast recovery from transient faults potentially challenging |
| Integrated HW Safety Architecture | Feasible  
  • However, redundancy on component level is typically a technical overkill  
  • Functional safety enablement simplifies demonstration of compliance |
| | | Feasible  
  • Integrated safety architecture using dual-core lockstep uC  
  • Functional and temporal alignment between two channels simplified  
  • Fast recovery from transient faults more feasible |
Tradeoffs of Different Redundancy Approaches

**Algorithm related approaches**
- different modules
- different submodules
- different FFs
- plausibility check
- different math/flow
- main prog & float & integer

**Time related approaches**
- different clock cycles
- concurrent threads
- one after another

**HW related approaches**
- different chips
- different die areas
- different modules
- different submodules

⇒ Tradeoff: SW Complexity
⇒ Tradeoff: HW Complexity
⇒ Tradeoff: Performance
From System Level to Component Level

• Functional safety is not just an issue on system level but also on component level
  • Integrated safety devices (customer has no “direct” access to details of safety functions)

• Functional safety standards explicitly address component level
  • ISO 26262: Safety Element out of Context (SEooC)

• Basic approach is to assume a system context (or several) of the component
  • Safety Application Guide (Safety Manual) specifies how the component is applied correctly in the assumed system context
How does the system context impact safety?

Customer asks for a “safe ladder”

The “safe ladder” in the field
History of Auto MCU Functional Safety Solutions

- **Gen 1 Safety** More than 10 years experience of safety development in the area of MCU & SBC
- **Gen 2 Safety** First general market MCU, MPC5643L ➞ Certified ISO 26262!
- **Gen 3 Safety** From 2012, multiple MCUs in Body, Chassis and Powertrain are being designed and developed according to ISO 26262

**Functional Safety Solutions**

- **2000**
  - **Custom IC**
    - Started to ship in 2000 first safe MCU for braking applications
    - IEC 61508 / ISO 26262 compliance achieved at system level (top down approach)
    - MCU features are a key enabler for SIL3 / ASILD

- **2008**
  - **Custom Safety Platform for Braking**
    - 32-bit Dual-Core MCU
    - Developed according to ISO 26262
    - Target Applications for Chassis – ASILD

- **2012**
  - **Gen 3 Safety**
    - MPC5744P/MPC5777K/etc 55nm
      - 32-bit Dual/Quad-Core MCU
      - Developed according to ISO 26262
      - Target Applications Chassis & P/T for – ASILD
      - Safe methodology, Architecture, SW and tools
  - **PowerSBC Gen 2**
    - Voltage Supervision
    - Fail-Safe State Machine
    - Fail-Safe IO
    - Advanced Watchdog

- **PowerSBC**
  - Voltage Supervision
  - Fail-Safe State Machine
  - Fail-Safe IO
  - Advanced Watchdog

= 32-bit Dual-Core MCU
Our Vision in Safety: Know Your Context

Ideal partitioning between HW and SW measures dependant on ASIL target and complexity of safety function

Example Safety Applications …

High Redundancy

Redundancy of application data

Simple Safety Function

Complex Safety Function

ASIL D target

ASIL C target

ASIL B target

ASIL A target

Functional Safety Effort

Example:

ASIL D = ASIL A + ASIL C

Decomposition to reduce complexity of single instances
Our solution for safety: Balance and flexible

Offering products that scale to application specific safety requirements

**simple** Safety Function

**ASIL D** target

**ASIL C** target

**ASIL B** target

**ASIL A** target

**complex** Safety Function

Functional Safety Effort

Spanning the whole range efficiently …
## SafeAssure Products

<table>
<thead>
<tr>
<th>Target Market</th>
<th>Product Type</th>
<th>Product</th>
<th>Target Applications</th>
<th>Safety Process</th>
<th>Safety Hardware</th>
<th>Safety Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automotive</td>
<td>MCU</td>
<td>MPC5748M</td>
<td>Diesel Engine Management</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Manual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPC574BF</td>
<td>Electric Power Steering</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Manual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPC567xK</td>
<td>77 GHz RADAR System</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Application Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPC564xL</td>
<td>77 GHz RADAR System</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Manual, System Level Application Note, Safety Application Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPC560xP</td>
<td>Dual Airbag System</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Single core, SEC/DEEC ECC, Clock Monitoring Unit, Low Voltage Detector, FOCU</td>
<td>FMEDA, Safety Application Note</td>
</tr>
<tr>
<td>Industrial</td>
<td>MCU</td>
<td>MX3100</td>
<td>Aerospace</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Application Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PS520</td>
<td>Process Control, Temperature Control</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Application Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PS530</td>
<td>Programmable Logic Control (PLC)</td>
<td>ISO/TS 26262 ASIL D</td>
<td>Integrated Safety Architecture e.g.: Dual core, isolated or dual parallel processing, replicated peripherals, FOCU</td>
<td>FMEDA, Safety Application Note</td>
</tr>
</tbody>
</table>

To view the latest SafeAssure product table visit www.freescale.com/SafeAssure
Safety Concept of an MCU with Integrated Safety Architecture
Random Failures and their Handling

- **Single Point Failure (SPF)**
  - Immediate potential to cause a hazard
  - Quick detection or mitigation

- **Latent Failure (LF)**
  - Can become dangerous in conjunction with a second fault
  - Can aggregate
  - Periodic detection

- **Common Cause Failure (CCF)**
  - Causes several component to fail
  - Can possibly annul redundancy-based measures
  - Mitigation or quick detection
MCU Countermeasures for Failure Classes

- **Single Point Failure (SPF)**
  - Structural redundancy
    - Core, DMA
  - Information redundancy
    - **E2E ECC**, EDC on Cache

- **Latent Failure (LF)**
  - HW-Self test
    - Memory, logic
    - 90% stuck-at

- **Common Cause Failure (CCF)**
  - Delayed Checker Core
  - Supervision of clock, power and temperature
  - Independent safety clock
  - Independent failure signaling
Safety Concept Example: MPC5643L (1)

- Target applications
  - Safety applications that require a high safety integrity level, such as:
    - Electric power steering
    - Electronic stability control
- Item must be in safe state for modes of (non)operation:
  - Completely unpowered
  - Reset
  - Operating correctly
  - Indicating an internal error
- Safety mechanism: technical solution to detect faults or control failures in order to achieve a safe state
Safety Concept Example: MPC5643L (2)

- Safety mechanisms:
  - Built-in self tests (memory, logic, ADC)
  - Duplicate computational elements in lock-step
  - ECC for FLASH/SRAM
  - Temperature, clock and voltage monitors
  - Fault Collection and Control Unit (FCCU) with redundant fault notification path
  - Independent safety clock
  - eDMA and CRC
  - Access protection (MPU, register) ....
Safety Concept Example: MPC5643L (3)

- Elements having *low* application dependency
  - Safety architecture may not interfere with application
  - Hardware driven

- Elements having *high* application dependency
  - Functional safety of the periphery is ensured by system-level measures
  - Flexible usage within application software
First ISO 26262 Certified MCU—Qorivva MPC5643L

- Certified by exida – an independent accredited assessor
- Certificate issued based on a successful assessment of the product design and applied development and production processes against all requirements and work product definitions of ISO 26262 identified as applicable to an MCU part
- MPC5643L MCU certified for use for all Automotive Safety Integrity Levels (ASIL), up to and including the most stringent level, ASIL D

Released on 6th September, 2012
ISO 26262 Assessment and Audit Summary

- Assessment of the MPC5643L Safety Case
- Assessment and audit of Freescale’s development processes used for the MPC5643L
- Assessment of the FMEDA (Failure Modes Effects and Diagnostic Analysis) of the MPC5643L to confirm it satisfies the SPFM, LFM and PMHF metrics required for ASIL D
- Assessment of the MPC5643L hardware design, implementation and verification activities
- Over 50 work products were provided to exida during the assessment and on-site audits
Safety Software
Freescale Automotive Software

- Freescale Automotive Software is mostly focused on AUTOSAR MCAL and OS.

AUTOSAR Runtime Environment (RTE)

Application Layer

Not all modules are shown here.
Freescale Automotive Software

However, also about
- Instruction-based core self-test
- Libraries such as math library, motor control libraries, etc.
- Complex drivers such as Pulse Width Modulator (PWM) and Ethernet

ISO26262 imposes that all hardware and software elements are designed and developed to minimize the risk of causing hazardous events.

Freescale software for SafeAssure meets ISO26262
- supports hardware to meet ISO26262 requirements
  - detection of HW random faults
- supports efficient achievement of safety goals
  - detection of SW systematic faults
    - assuring freedom from interference or preventing interference
    - following ISO26262 compliant FSL SW development process
- reaction to faults
... Toward Functional Safety Software Portfolio

- Ordinary Software Offering
  - MCAL
  - OS
  - ICST
  - MCLib (Beta)

- SafeAssure Software Offering
  - sMCAL
  - sOS
  - sCST
  - sMCLib
  - sPTLib
  - SafeLib
  - ...

---

34
Objectives of SafeAssure Software Portfolio

• Support efficient achievement of safety goals up to ASIL-D
  – Safety with minimized performance degradation
  – Safety simplified for integrators
  – Cross-platform consistent architecture

• Support achievement of hardware architectural metrics up to ASIL-D

All products in the Software SafeAssure portfolio are Safety Element out of Context (SEooC)

• safety-related requirements are assumed
• safety-related role is assumed
• deployment is envisioned
Software Portable Architecture for SafeAssure Solutions

- Safety-Related Functional Layer
- Safety Service Layer
- HW Safety Layer

Software components that facilitate and support safety-related applications.

Software components for detecting hardware faults to support compliance with ISO26262 hardware architectural metrics:
- SPFM
- LFM

µPs with different set of safety measures and safety support functions

Software functional components that may carry out safety-related functions
## Classes of Freescale Software Components

<table>
<thead>
<tr>
<th>Freescale Software Product Class</th>
<th>Products</th>
</tr>
</thead>
</table>
| Safety-Related Functional Components | • safety MCAL (sMCAL)  
• safety Motor Control Lib (sMCLib) |
| Safety Service Components | • Safety Library (SafeLib)  
  • Microcontroller Error Management  
    • Software support for FCCU, MEMU, LBIST, MBIST  
    • Hardware error collection  
  • Safety Error Reporting and Reaction  
    • Collect both Hardware and Software faults  
    • Provides reaction mechanisms  
  • Resource Manager  
    • Manages peripheral control to enable run-time invocation of peripheral tests  
  • CRC driver  
    • Abstracts HW/SW implementation  
  • DMA protection  
  • Software Integrity Universal Checker  
  • … tbd  
  • safety Operating System (sOS)  
    • FSL sOS / external sOS |
| HW Safety Components | • safety Core Self Test (sCST)  
• safety Peripheral Test Library (sPTLib) |
Safety Support
Failure Mode, Effect and Diagnostic Analysis

- A systematic way to identify and evaluate failure modes, effects, and diagnostic techniques, and to document the system.
- Target values can be assigned to MCU
- FMEDA for MCU => for system-level FMEDA

Element
gate count / size

Failure Mode /
Distribution

Technology
dependent
failure rate

Safety Mechanism

FMEDA

Raw failure
rates

PMHF

SPFM

LFM

ASIL D targets for whole item

Typical results for MCU with integr. safety architecture
SEooC safety assumption can be inflexible

Old SEooC assumption forces system vendors to implement unnecessary safety measures!

- Example: First generation safety application guide for had 68 mandatory requirements
Without application context, an SEooC analysis requires safety measures for:

- PWM
- ADC
- SPI
- Core Subsystem

Consequence: over-engineering of architecture and suboptimal partitioning of software and hardware effort often
A new approach shall complement the safety measures of a context, not duplicate what the context already provides!
FMEDA Example: MPC5643L

- FMEDA for MPC5643L
  - Processing units (core, etc.)
  - Power supply
  - Clock
  - Non-volatile memory (FLASH)
  - Volatile memory (SRAM)

- Safety concept on system level not known
  => Raw failure rates for
  - Digital I/O
  - Analogue I/O
  - External communication
Dynamic FMEDA

Freescale introduces dynamic FMEDA approach:

- Customer communicates implemented safety measures and Freescale delivers respective tailored FMEDA (within e.g. 1 hour)
- E.g. MPC5675K has more than >1 million different FMEDAs in database – so truly back to the world of general purpose!
- No longer applications have to fulfill FMEDA assumption but FMEDA tailors to application
### Flexible FMEDA

<table>
<thead>
<tr>
<th>Software Functional Self Test Routine for Core supported by Hardware periodically executed within Fault Tolerant Time Interval</th>
<th>Lockstep enabled</th>
<th>Safety Relevant Core 2 Usage</th>
<th>Temporal Core and DMA Redundancy (recalculate on same core or double move with same DMA)</th>
<th>Window and Logical Monitoring Watchdog implemented and detecting failure within Fault Tolerant Time Interval</th>
<th>MPU Enabled</th>
<th>MNUEnabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>Diagnostic Coverage of Self Test Routine</td>
<td>Reciprocal comparison</td>
<td>Window Monitoring Watchdog configured</td>
<td>Logical Monitoring Watchdog configured</td>
<td>Replicating Software use different SRAM block</td>
<td>50% diagnostic coverage</td>
<td></td>
</tr>
<tr>
<td>30% diagnostic coverage</td>
<td>100% diagnostic coverage</td>
<td></td>
<td></td>
<td>Reciprocal comparison within Fault Tolerant Time</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>Software Test within Fault Tolerant Time Interval</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>Software Test supported by hardware</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>50% diagnostic coverage</td>
<td>50% diagnostic coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Target Achievement respective to ISO 26262 and IEC 61508 Ed. 2.0**

| Single-Point Fault Metric | ≥ 99.84% | ASIL D requires a Single-Point fault Metric ≥ 99% |
| Latent Fault Metric | ≥ 99.94% | ASIL D requires a Latent Fault Metric ≥ 90% |
| SFF | ≥ 99.84% | SIL3 requires a Single-Point fault Metric ≥ 99% |

\[
\lambda_{SPF} + \lambda_{RF} (ISO26262), \lambda_{DU} (IEC61508) = 2.18E-10 \text{ h}^{-1} \quad \text{ASIL D & SIL3 requires a single point or dangerous undetected failure rate of ≤ 1E-9} \\
\lambda_{total,ISO26262} = 1.38E-07 \text{ h}^{-1} \quad \text{ASIL D & SIL3 requires a single point or dangerous undetected failure rate of ≤ 1E-9} \\
\lambda_{total,IEC61508} = 1.38E-07 \text{ h}^{-1} 
\]
Safety Support – FMEDA, Documentation & More

**FSL QM Products - Typical Deliverables**

- Safety Analysis of Architecture: Safety FMEA or FTA
- User Guide: Safety Application Note
- Development Process evidence: PPAP, Quality Plan (Mapping to ISO 26262 / IEC 61508 checklists)

**ISO 26262 or IEC 61508 Products – Typical Deliverables**

- Safety Analysis of Architecture: FMEDA, CCA or FTA
- Development Process evidence: PPAP, Safety Plan, Certificates

**Local Support**

- Functional Safety Field Experts

**Learning**

- Field Training / workshops – delivered by Local Functional Safety FAE Experts
Safety Support – Dynamic FMEDA

Objective
- Tailor FMEDA to match application configuration
- Enables customers, by supporting their system level architectural choices

Content
- FMEDA methods aligned with functional safety standards
  - SPFM & LFM, PMFH – ISO 26262
  - SFF & PFH- IEC 61508 ed-2.0
  - βic – IEC 61508 ed-2.0 part 2, Annex E
- Dynamic FMEDA covers elements with low application dependency: Clock, Power Supply, Flash, STM, SRAM, Processing Unit…

Work flow and result
- Customer specifies the Safety Integrity Level required by their application, and then confirms the Safety Measures that will be used
- A tailored FMEDA is then supplied to customer’s for their specific application

Objective
- Enables customers to extract the full value of Freescale's functional safety offering
- Simplify integration of Freescale's safety products into applications
- A comprehensible description of all information relating to FS in a single entity to ensure integrity of information and links with datasheet

Content
- SoC Safety Concept description
- System level assumptions of use (Safety specific usage considerations)
- Pseudo-code or C-Code to simplify adoption of safety application requirements
- FMEDA results
  - Latent Fault Matrix (LFM)
  - Single Point Fault Matrix (SPFM)
  - Probabilistic Metric for random Hardware Failures (PMHF)
- Provisions against Dependent Failures

Safety Manual for MCU Solution
- Freescale Semiconductor Safety Manual

Safety Manual for MPC574xP
- Safety Manual for Analog Solution
- Freescale Semiconductor Safety Manual

Safety Manual for MC33906/7/8
- Freescale Semiconductor Safety Manual
Design Guidelines for

- Integration of Microcontroller and Analog & Power Management device
- Explains main individual product Safety features
- Uses a typical Electrical Power steering application to explain product alignment
- Covers the ASIL D safety requirements that are satisfied by using both products:
  - MPC5643L requires external measures to support a system level ASIL D safety level
  - MC33907/08 provides those external measures:
    - External power supply and monitor
    - External watchdog timer
    - Error output monitor

Integrating the MPC5643L and MC33907/08 for ISO26262 ASIL-D Applications

This application note provides design guidelines for integrating the Freescale MPC5643L microcontroller unit (MCU) and Freescale MC33907/08 System Beats Chip in automotive electronic systems targeting the ISO 26262 functional safety standard. It provides an overview of the MPC5643L and the MC33907/08 features set and covers the functional safety requirements that are satisfied in order to achieve ASIL D level of safety.

Integrating the MPC5643L and MC33907/08 in a system provides many advantages for the customer. Freescale’s ISO 26262 solutions, that form part of the Freescale SafeAssure program, help system manufacturers more easily achieve system compliance with functional safety standards by simplifying the system architecture.

I. MPC5643L Overview

This section describes the MPC5643L features that are of interest when integrating the device with the MC33907/08.

A. Safety Concept

The MPC5643L is a fully configurable 32-bit PowerPC-440 core, a processor for SoC (System on Chip) safety platform with a safety concept targeting ISO 26262 ASIL-D integrity level. In order to minimize additional software and module level features to reach this target, on-chip redundancy is offered for the critical components of the MCU (CPU core, DMA controller, interrupt controller, processor bus system, memory protection unit, flash memory and RAM controllers, peripheral bus bridges, systems timers, and system watchdog timer). A Redundancy control and checker unit (RCCU) is implemented at each output of this SoC. RCCU is available for on chip RAM and flash memories. The Programmable Fault Collection and Control Unit (PFCU) monitors the integrity status of the device and provides flexible safe state control.

B. Power Supply Requirements

The on-chip voltage regulator module provides the following features: Single high supply requires nominal 2.5V. An external power transistor is used to reduce disconnection capacity at high temperature, but if high temperature is maintained within package dissipation capacity (lower frequency of operation). All I/Os are at same voltage.
Summary

• ISO 26262 has been widely adopted for Automotive functional safety

• Systems with safety goals according to ISO 26262 are no longer an exception

• Freescale support OEMs and Tier1s to achieve their ISO 26262 safety goals
  - Discrete and integrated safety architectures
  - System level chip set solutions – beyond MCU

• Combined with a range of Safety HW products, Freescale supports customers by providing a set of differentiating collateral that enable our customers and can significantly reduce their development time
  - Dynamic FMEDA, Safety Manual, System Level Application Notes