This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie-Sklodowska-Curie grant agreement number 721624.
Introduction to Functional Safety

RADSAGA Training Workshop – March 2018
Outline

• Autonomous Driving Challenges
• Functional Safety
• Structure of ISO 26262
• Safety Analysis
• Random failures
Introduction

Functional safety is not immune to conflicts.....

- top-down vs bottom-up
- S/W vs H/W failures
- statistical vs formal verification
- gurus vs automation

...and, as in most of the conflicts, both parties have a piece of the solution in their hands.....
<table>
<thead>
<tr>
<th>SAE Level</th>
<th>Name</th>
<th>Narrative Definition</th>
<th>Execution of Steering and Acceleration/Deceleration</th>
<th>Monitoring of Driving Environment</th>
<th>Fallback Performance of Dynamic Driving Task</th>
<th>System Capability (Driving Modes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Automation</td>
<td>The full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems</td>
<td>Human Driver</td>
<td>Human Driver</td>
<td>Human Driver</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>Driver Assistance</td>
<td>The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task</td>
<td>Human Driver and System</td>
<td>Human Driver</td>
<td>Human Driver</td>
<td>Some Driving Modes</td>
</tr>
<tr>
<td>2</td>
<td>Partial Automation</td>
<td>The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task</td>
<td>System</td>
<td>Human Driver</td>
<td>Human Driver</td>
<td>Some Driving Modes</td>
</tr>
</tbody>
</table>
### The Autonomous Driving levels (SAE) – tomorrow

<table>
<thead>
<tr>
<th>SAE Level</th>
<th>Name</th>
<th>Narrative Definition</th>
<th>Execution of Steering and Acceleration/Deceleration</th>
<th>Monitoring of Driving Environment</th>
<th>Fallback Performance of Dynamic Driving Task</th>
<th>System Capability (Driving Modes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Conditional Automation</td>
<td>The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene</td>
<td>System</td>
<td>System</td>
<td>Human Driver</td>
<td>Some Driving Modes</td>
</tr>
<tr>
<td>4</td>
<td>High Automation</td>
<td>The driving mode-specific performance by an automated driving system of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene</td>
<td>System</td>
<td>System</td>
<td>System</td>
<td>Some Driving Modes</td>
</tr>
<tr>
<td>5</td>
<td>Full Automation</td>
<td>The full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver</td>
<td>System</td>
<td>System</td>
<td>System</td>
<td>All Driving Modes</td>
</tr>
</tbody>
</table>
The essence of an Autonomous Driving platform

Intelligent eyes

- Vision.

Intelligent & powerful brain

- Perception and fusion.
- Modeling and planning.
- Decision making.

We need high-performance computing, flexibility, and programmability.
Autonomous Driving is E2E
Autonomous Driving is powered by data

<table>
<thead>
<tr>
<th>Category</th>
<th>Data Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Internet User</td>
<td>~1.5 GB of traffic per day</td>
</tr>
<tr>
<td>Smart Hospital</td>
<td>3,000 GB per day</td>
</tr>
<tr>
<td>Autonomous Vehicles</td>
<td>4,000 GB per day... each</td>
</tr>
<tr>
<td>Airplane Data</td>
<td>40,000 GB per day</td>
</tr>
<tr>
<td>Smart Factory</td>
<td>1,000,000 GB per day</td>
</tr>
</tbody>
</table>
...can we trust Autonomous Driving?

H/W and S/W complexity is expected to grow at least by a factor 20 in the next few years, so higher risk of failures....

Connectivity brings security threats....

...and autonomous systems are expected to detect & control failures!

I’m sorry Dave. I’m afraid I CAN’T do that.
The challenges of Autonomous Driving

H/W and S/W complexity is expected to grow at least by a factor 20 in the next few years, so higher risk of failures....

Connectivity brings security threats....

...and autonomous systems are expected to detect & control failures!
T patient, Autonomous Driving

Safety of Intended Functionality (SOTIF)

Anomaly detection and reaction

Functional Safety for H/W and S/W

Security
Functional Safety

The absence of unreasonable risk due to hazards caused by malfunctioning behaviour of E/E systems

Systematic failures
(Bugs in S/W, H/W design and Tools)

Random H/W failures
(permanent faults, transient faults occurring while using the system)

Ruled by International Standards
setting the "state of art" (for liability)
Reducing the risk

Aim of functional safety is therefore reducing the risk of harm from unacceptable to tolerable level.

- “Tolerable” is defined according corporate rules, government regulations, law, public opinion....

In ISO 26262:

- risk = combination of the probability of occurrence of harm and the severity of that harm
  - In ISO 26262, occurrence is a function of exposure and controllability

- harm = physical injury or damage to the health of persons
  - ISO 26262 scope: the users and occupants of the vehicle, other traffic participants and other endangered/exposed persons

Completely removing the risk is NOT possible
Risk evaluation

- Probability per hour (runtime)
- Severity of possible accident
- Probability of exposure to driving situation where accident can potentially happen
- Risk Reduction external to technical system: e.g. driver controls situation

Aim of functional safety

acceptable
not acceptable

Tolerable Risk

Lower than tolerable risk
Residual Risk

Extreme improbable
Very rarely
Rarely
Sometimes
Always

Low
Important
Hazardous
(Catastrophically)

Probability of exposure to driving situation where accident can potentially happen
Risk Reduction external to technical system: e.g. driver controls situation

Aim of functional safety

- Probability per hour (runtime)
- Severity of possible accident
- Probability of exposure to driving situation where accident can potentially happen
- Risk Reduction external to technical system: e.g. driver controls situation

Aim of functional safety

acceptable
not acceptable

Tolerable Risk

Lower than tolerable risk
Residual Risk

Extreme improbable
Very rarely
Rarely
Sometimes
Always

Low
Important
Hazardous
(Catastrophically)
ISO 26262 2nd edition
ISO 26262 today

ISO 26262 parts:

- part 1, Vocabulary;
- part 2, Management of functional safety;
- part 3, Concept phase;
- part 4, Product development: system level;
- part 5, Product development: HW level;
- part 6, Product development: SW level;
- part 7, Product operation, service and decommissioning;
- part 8, Supporting processes;
- part 9, ASIL-oriented and safety-oriented analyses;
- part 10, Guideline
- part 11, Guideline for Semiconductors
- part 12, Adaption for motorcycles
ISO 26262 and law

Do you have to fulfill ISO 26262 by law?

- NO

However, if after a car accident you are in a Court of Law, you will be asked...

- Did you follow “state of the art”?

Functional safety standards are considered by law the minimum level of the “state of the art”

So you have to fulfill ISO 26262 to avoid troubles....
ISO26262 Overview

Provides an automotive safety lifecycle (management, development, production, operation, service, decommissioning) and supports tailoring the necessary activities during these lifecycle phases.

Covers functional safety aspects of the entire development process (including such activities as requirements specification, design, implementation, integration, verification, validation, and configuration).

Provides an automotive-specific risk-based approach for determining risk classes (Automotive Safety Integrity Levels, ASILs).

Uses ASILs for specifying the item’s necessary safety requirements for achieving an acceptable residual risk.

Provides requirements for validation and confirmation measures to ensure a sufficient and acceptable level of safety is being achieved.
About hazards

Hazard:
- potential source of harm caused by malfunctioning behavior of the item

ISO 26262 addresses:
- possible hazards caused by malfunctioning behavior of E/E safety-related systems including interaction of these systems.

It does not address hazards related to:
- electric shock, fire, smoke, heat, radiation, toxicity, flammability, reactivity, corrosion, release of energy, and similar hazards unless directly caused by malfunctioning behavior of E/E safety-related systems.

ISO 26262 does not address the nominal performance of E/E systems
Fault, Error, Failure

**fault:**
- abnormal condition that can cause an element or an item to fail

**error:**
- discrepancy between a computed, observed or measured value or condition, and the true, specified, or theoretically correct value or condition

**failure:**
- termination of the ability of an element, to perform a function as required
Failure classes

random hardware failure:
- failure that can occur unpredictably during the lifetime of a hardware element and that follows a probability distribution

systematic failure:
- failure, related in a deterministic way to a certain cause, that can only be eliminated by a change of the design or of the manufacturing process, operational procedures, documentation or other relevant factors
Achieving the "ASIL"

Achieving Automotive Safety Integrity

- Setting up functional safety management
- Defining the safety goal

Improving the process

Avoid systematic failures

Improving the product

Detect / Tolerate
HW Random Failures

Avoid / Detect
Dependent failures
Functional safety management

It is a set of requirements related to how functional safety shall be planned, executed, documented and verified during the product lifecycle.....

....it’s a bit boring part....

...but is an important part to learn about how to avoid or detect SYSTEMATIC failures.....
The ISO26262 Lifecycle
Confirmation measures

Functional Safety Assessment

Confirmation measures
„ASIL“ from top or from bottom?

- Achieving Automotive Safety Integrity
  - Setting up functional safety management
  - Defining the safety goal
  - Improving the process
    - Avoid systematic failures
  - Improving the product
    - Detect / Tolerate HW Random Failures
    - Avoid / Detect Dependent failures
Top-down

Functional safety applies to Electric, Electronic, Programmable Electronic systems (E/E/PE) systems carrying out safety functions.

Top-down approach (application dependent)
ISO 26262 introduced also a bottom-up approach, i.e. elements developed based on assumed requirements.

**Bottom-up approach** (application independent)

The safety element “out-of-context”
The SEooC „definition”

A Safety Element out of Context (SEooC) is a safety-related element not developed for a specific item.

- An SEooC can be a system, subsystem, a software component, or a hardware component or hardware parts.
- An SEooC is never an item, i.e. if it is a system, this system is not integrated into the context of a vehicle and so it is not an item.

Examples of SEooC:
- Airbag Control System, Start Stop System, Engine management controller, microcontrollers, SW implementing a communication protocol, AUTOSAR application SW modules, AUTOSAR basic SW module etc...
The risk/hazard analysis

1. Identify the hazards

2. Identify the operational situations

3. Link hazards with operational situations (i.e. build hazardous events)
Risk quantification

RISK ASSESSMENT

Tolerable Risk = Severity x Occurrence

Exposure x Controllability x Failure rate

ASIL: Automotive Safety Integrity Level
# Severity

The potential severity of the resulting harm or damage

<table>
<thead>
<tr>
<th>Class</th>
<th>S0</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>No injuries</td>
<td>light and moderate injuries</td>
<td>Severe injuries, possibly life-threatening, survival probable</td>
<td>Life-threatening injuries (survival uncertain) or fatal injuries</td>
</tr>
<tr>
<td><strong>Reference for single injuries (from AIS scale)</strong></td>
<td>AIS 0 and less than 10% probability of AIS 1-6 Damage that cannot be classified safety-related</td>
<td>more than 10% probability of AIS 1-6 (and not S2 or S3)</td>
<td>more than 10% probability of AIS 3-6 (and not S3)</td>
<td>more than 10% probability of AIS 5-6</td>
</tr>
<tr>
<td><strong>Informative examples</strong></td>
<td>Bumps with roadside infrastructure Pushing over roadside post, fence, etc. Light collision Light grazing damage Etc...</td>
<td>Side impact with a narrow stationary object, e.g. crashing into a tree (impact to passenger cell) with <strong>very low speed</strong> Etc...</td>
<td>Side impact with a narrow stationary object, e.g. crashing into a tree (impact to passenger cell) with <strong>low speed</strong> Etc...</td>
<td>Side impact with a narrow stationary object, e.g. crashing into a tree (impact to passenger cell) with <strong>medium speed</strong> Etc...</td>
</tr>
</tbody>
</table>

AIS = Abbreviated Injury Scale
Exposure

how frequently and for how long individuals are in a situation where the hazardous event can occur

<table>
<thead>
<tr>
<th>Class</th>
<th>Temporal Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E1</td>
</tr>
<tr>
<td>Description</td>
<td>Very low probability</td>
</tr>
<tr>
<td>Definition</td>
<td>Not specified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Road layout</th>
<th>Duration (% of average operating time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mountain pass with unsecured steep slope</td>
<td>Highway</td>
</tr>
<tr>
<td>One-way street (city street)</td>
<td></td>
</tr>
<tr>
<td>Country road intersection</td>
<td>Secondary Road</td>
</tr>
<tr>
<td>Highway entrance ramp</td>
<td>Country Road</td>
</tr>
<tr>
<td>Highway exit ramp</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Road surface</th>
<th>Duration (% of average operating time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snow and ice on road</td>
<td>Wet road</td>
</tr>
<tr>
<td>Slippery leaves on road</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nearby elements</th>
<th>Duration (% of average operating time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lost cargo or obstacle</td>
<td>Highway</td>
</tr>
<tr>
<td>obstacle in lane of travel</td>
<td></td>
</tr>
</tbody>
</table>

Etc...
Controllability

the ability of the avoidance of specific harm or damage through timely reactions of the persons involved

<table>
<thead>
<tr>
<th>Class</th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Controllable in general</td>
<td>Simply Controllable</td>
<td>Normally Controllable</td>
<td>Difficult to Control or Uncontrollable</td>
</tr>
<tr>
<td>Driving Factors &amp; Scenarios</td>
<td>Controllable in general</td>
<td>99% or more of all drivers or other traffic participants are usually able to avoid harm</td>
<td>90% or more of all drivers or other traffic participants are usually able to avoid harm</td>
<td>Less than 90% of all drivers or other traffic participants are usually able, or barely able, to avoid harm</td>
</tr>
<tr>
<td>Unexpected radio volume increase</td>
<td>Maintain intended driving path</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blocked steering column</td>
<td></td>
<td>Brake to slow/stop vehicle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failure of ABS during emergency braking</td>
<td></td>
<td>Maintain intended driving path</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty driver airbag release when travelling at high speed</td>
<td></td>
<td></td>
<td>Maintain intended driving path, stay in lane Brake to slow/stop vehicle</td>
<td></td>
</tr>
</tbody>
</table>

For C2, 20 valid data sets per scenario may be required
# ASIL Determination

<table>
<thead>
<tr>
<th>Severity class</th>
<th>Exposure class</th>
<th>Controllability class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>C1</td>
</tr>
<tr>
<td>S1</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>QM</td>
</tr>
<tr>
<td>S2</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>A</td>
</tr>
<tr>
<td>S3</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>B</td>
</tr>
</tbody>
</table>
**Allocation of requirements**

The functional safety requirements shall be allocated to architectural elements

- the set of elements that have safety requirements allocated to them and their interaction is referred as "**safety architecture**"

- The safety architecture includes the redundancy and the independence concept for the elements in block diagrams form

---

**Results of hazard analysis and risk assessment**

- **Safety goal A**
  - ASIL 3-7
- **Safety goal B**
  - ASIL 3-7
- **Safety goal N**
  - ASIL 3-7

**Functional safety requirement**

- **Assigned ASIL**
- **Allocated to subsystem**
Main workproducts (system)

Plans
- Validation plan
- Item integration and testing plan

Safety requirements
- Technical safety requirements specification
- System-level verification report

Design documents
- Technical safety concept
- System design specification
- Requirements for production/operation/service/decommissioning
- Hardware-Software Interface Specification (HSI)

Testing and verification
- Integration testing specification and related testing reports
- Validation report
- Functional safety assessment report
- Release for production report
Safety Analysis
Safety analyses

The scope of the safety analyses includes:

- the validation of safety goals and safety concepts;
- the verification of safety concepts and safety requirements;
- the identification of conditions and causes, including faults and failures, that could lead to the violation of a safety goal or safety requirement;
- the identification of additional requirements for detection of faults or failures;
- the determination of the required responses (actions/measures) to detected faults or failures; and
- the identification of additional requirements for verifying that the safety goals or safety requirements are complied with, including safety-related vehicle testing.
Qualitative vs Quantitative

Safety analyses are performed at different level of abstraction during the lifecycle.

*Qualitative* analysis methods identify failures but do not predict the frequency of failures.

*Quantitative* analysis methods are used to predict the frequency of failures. They are:

- complement qualitative analyses
- are used to verify a HW against defined targets for avoidance of violation of a safety goal due to hardware random failures.
- require additional knowledge of quantitative failure rates of hardware elements and their combinations
- address random failures of hardware only. They are not applied neither to systematic failures nor to dependent failures.
FMEA and FTA

- Deductive and inductive analyses are used at system level to identify causes and effects of systematic failures
  - Deductive analysis methods include FTA, reliability block diagrams, Ishikawa diagram
  - Inductive analysis methods include FMEA, ETA, Markov modeling

<table>
<thead>
<tr>
<th></th>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>Deductive analysis</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Inductive analysis</td>
<td>++</td>
</tr>
</tbody>
</table>
FMEA vs FTA

FMEA

- Final Effects (Failures)
  - Intermediate Effects (Errors)
    - Elementary Root Causes (Faults)

FTA

- Final Effect (Failure)
  - Intermediate Effects (Errors)
    - Elementary Root Causes (Faults)
      - Not Covered

RADSAGA Training Workshop
Systematic failures avoidance

- In order to avoid failures resulting from high complexity, the architectural design shall exhibit
  - modularity
  - adequate level of granularity

<table>
<thead>
<tr>
<th>Properties</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>Hierarchical design</td>
<td>+</td>
</tr>
<tr>
<td>Precisely defined interfaces</td>
<td>+</td>
</tr>
<tr>
<td>Avoidance of unnecessary complexity of hardware components and</td>
<td>+</td>
</tr>
<tr>
<td>software components</td>
<td></td>
</tr>
<tr>
<td>Avoidance of unnecessary complexity of interfaces</td>
<td>+</td>
</tr>
<tr>
<td>Maintainability during service</td>
<td>+</td>
</tr>
<tr>
<td>Testability during development and operation</td>
<td>+</td>
</tr>
</tbody>
</table>

RADSAGA Training Workshop
About fault models

Random

HW

Permanent

Intermittent

Transient

d.c. faults
- stuck-at
- stuck-open
shorts

SEE
- SBU
- MBU
- SET
Failure rates sources

The estimated failure rates for hardware parts used in the analyses shall be determined either:

- Using hardware part failure rates data from a recognized industry source
- Using statistics based on field returns or tests
  - the estimated failure rate should have an adequate confidence level
- Using expert judgment founded on engineering approach based on quantitative and qualitative arguments.

From component’s providers those data might be based on number of (random) failures divided by equivalent device hours.

- These are obtained from field data or from accelerated life testing (as defined in standards such as JEDEC and AEC) scaled to a mission profile (e.g. temperature, on/off periods) with the assumption of a constant failure rate (random failures, exponential distribution). The numbers are usually provided as a maximum FIT based on a sampling statistics confidence level.
The IEC 62380 (cont.)

MATHEMATICAL MODEL:

\[ \lambda = \left( \lambda_1 \times N \times e^{-0.35\alpha} + \lambda_2 \right) \times \frac{\sum_{i=1}^{N} (\pi_i) \times \tau_i}{\tau_{on} + \tau_{off}} \times \left( 2.75 \times 10^{-3} \times \pi_0 \times \left( \sum_{i=1}^{N} (\pi_i) \times (\Delta T_i)^{0.68} \right) \times \lambda_3 \right) + \left( \frac{\pi_j \times \lambda_{EOS}}{\lambda_{const}} \right) \times 10^{-9} / h \]

- Related to the type of part
- Related to temperature
- Related to the die area and technology type
- Related to package
- Related to package
- Related to overstress
### The IEC 62380 (cont.)

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silicon: MOS</td>
<td>Standard Circuits (3)</td>
</tr>
<tr>
<td>ABBREVIATIONS</td>
<td>TYPES</td>
</tr>
<tr>
<td>ROM</td>
<td>Read only memory</td>
</tr>
<tr>
<td>DRAM/VideoRAM/AudioRAM</td>
<td>Dynamic, Read Access Memory</td>
</tr>
<tr>
<td>High-speed SRAM, FIFO</td>
<td>Static Read Access Memory - First in First out register, (“mixed MOS”)</td>
</tr>
<tr>
<td>Low-consumption SRAM</td>
<td>Static Read Access Memory - Low consumption; (CMOS)</td>
</tr>
<tr>
<td>Double access SRAM</td>
<td>Double Access Static RAM</td>
</tr>
<tr>
<td>EPROM, UV-ROM, REPROM</td>
<td>Electrically programmable, UV erasable - Read only memory</td>
</tr>
<tr>
<td>OTP</td>
<td>One-time programmable EPROM</td>
</tr>
<tr>
<td>FLASH</td>
<td>Electrically programmable and erasable (block) (1)</td>
</tr>
<tr>
<td>EEPROM, Flash EEPROM</td>
<td>Electrically programmable and erasable (word) (2)</td>
</tr>
<tr>
<td><strong>Note:</strong> (1) Whole memory array or blocks of words erasable. (2) Blocks of words or word erasable. (3) MOS include CMOS, HCMOS, NMOS, ... technologies.</td>
<td></td>
</tr>
</tbody>
</table>

### Silicon: MOS: ASIC circuits

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCA (RAM based)</td>
<td>USER PROGRAMMABLE LOGIC DEVICE:</td>
</tr>
<tr>
<td>Silicon: MOS</td>
<td>Standard cell, Full Custom</td>
</tr>
<tr>
<td></td>
<td>Gate Arrays</td>
</tr>
<tr>
<td></td>
<td>Logic Cell Array electrically configured by external memory</td>
</tr>
<tr>
<td></td>
<td>Electrically Programmable and erasable (AND/OR array)</td>
</tr>
<tr>
<td></td>
<td>Electrically Programmable (interconnected macrocells array) (2)</td>
</tr>
<tr>
<td></td>
<td>4 per gate</td>
</tr>
<tr>
<td></td>
<td>4 per gate</td>
</tr>
<tr>
<td></td>
<td>40 per gate (1)</td>
</tr>
<tr>
<td></td>
<td>3 per grid point</td>
</tr>
<tr>
<td></td>
<td>100 per macrocell</td>
</tr>
<tr>
<td><strong>Note:</strong> (1) or 4000 per macrocell; (2) EEPROM, EPROM, or Antifuse technologies.</td>
<td></td>
</tr>
</tbody>
</table>
The IEC 62380 (cont.)

- Temperature profiles

\[
\sum_{i=1}^{V} \left( \pi_t \right)_i \times \tau_i \\
\frac{\tau_{on} + \tau_{off}}{A = 3480; (E_a = 0.3 \text{ eV})}
\]

<table>
<thead>
<tr>
<th>Technological structure</th>
<th>Temperature factor ( \pi_t )</th>
</tr>
</thead>
</table>
| MOS BiCMOS (low voltage)| \[
\frac{A}{e^{\left( \frac{1}{328/273} \right) t_f}}
\] |

<table>
<thead>
<tr>
<th>Mission profile phases</th>
<th>Temp. 1</th>
<th>Temp. 2</th>
<th>Temp. 3</th>
<th>Ratios on/off</th>
<th>2 night starts</th>
<th>4 day light starts</th>
<th>Non used vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application types</td>
<td>((t_{on})_1^\circ C)</td>
<td>(\tau_1)</td>
<td>((t_{on})_2^\circ C)</td>
<td>(\tau_2)</td>
<td>((t_{on})_3^\circ C)</td>
<td>(\tau_3)</td>
<td>(\tau_{on})</td>
</tr>
<tr>
<td>Motor control</td>
<td>32</td>
<td>0.02</td>
<td>60</td>
<td>0.01</td>
<td>85</td>
<td>0.02</td>
<td>0.05</td>
</tr>
<tr>
<td>Passenger compartment</td>
<td>27</td>
<td>0.00</td>
<td>6</td>
<td>0.04</td>
<td>85</td>
<td>0.00</td>
<td>0.05</td>
</tr>
</tbody>
</table>
The failure rates battle...

SN 29500
About transient faults

- Alpha particle / neutron hitting a silicon device
- SEE: Single-event effect
  - SEU: Single-event upset (bit flip)
  - SET: Single-event transient (glitch)
- SEL: Single-event latch-up (permanent)
- Random noise
- Signal integrity problems (cross-talk)
- Other (e.g. power spikes)

Focus of ISO26262

- "soft-errors" according common understanding
- "soft-errors" according Wikipedia or other theory books (according some sources = transient faults)

~1000 FIT/Mbit at 90nm
<1 FIT/Mgate at 90nm 100MHz
Failure rates sources

The estimated failure rates for hardware parts used in the analyses shall be determined either:

- Using hardware part failure rates data from a recognized industry source

- Using statistics based on field returns or tests
  - the estimated failure rate should have an adequate confidence level

- Using expert judgment founded on engineering approach based on quantitative and qualitative arguments.
Failure rates scaling

If failure rates from multiple sources are combined, they shall be scaled using a scaling factor

- Scaling is possible if a rationale for the scaling factor between two failure rates sources is available

<table>
<thead>
<tr>
<th>Data source for Target Value</th>
<th>Table 6 9.4.2.1 a</th>
<th>Field data 9.4.2.1 b</th>
<th>Quantitative analysis 9.4.2.1 c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std. Database 8.4.3 a</td>
<td>$\lambda_{k,F_0} (1)$</td>
<td>$\lambda_{k,F_0} = \pi_{F_0\rightarrow F_3} \times \lambda_{k,F_0}$</td>
<td>$\lambda_{k,F_0}$</td>
</tr>
<tr>
<td>Statistics 5.4.3 b</td>
<td>$\lambda_{k,F_0} = \pi_{F_0\rightarrow F_3} \times \lambda_{k,F_0}$</td>
<td>$\lambda_{k,F_0}$</td>
<td>$\lambda_{k,F_0}$</td>
</tr>
<tr>
<td>Expert judgment 8.4.3 c</td>
<td>$\lambda_{k,F_0} = \pi_{F_0\rightarrow F_3} \times \lambda_{k,F_0}$</td>
<td>$\lambda_{k,F_0}$</td>
<td>$\lambda_{k,F_0}$</td>
</tr>
</tbody>
</table>

(1) For some types of hardware parts, different handbooks can give different estimates of the failure rate of the same type of hardware part. Therefore the scaling factor can be used to scale the failure rates of a hardware part using different handbooks.

(2) To have a consistent approach, failure rates have the same origin as the failure rates used in the calculation of the target value.
Fault classes

Failure modes of a HW element

Non-safety-related HW element

Safety-related HW element

Safe fault

Safe fault

Detected multiple point fault

Perceived multiple point fault

Latent multiple point fault

Residual fault / single point fault

Two definitions for “safe”

“critical” faults (dangerous faults in IEC 61508)
Graphical representation

- Safe faults
- Single Point or Residual faults
- Latent Multiple Point faults
- Detected Multiple Point faults
- Perceived Multiple Point faults
Aim of relative metrics

Be objectively assessable
- metrics are verifiable and precise enough to differentiate between different architectures

Make available ASIL dependent pass/fail criteria;

Reveal whether or not the coverage by the safety mechanisms is sufficient

Ensure robustness concerning uncertainty of hardware failures rates

Support usage on different elements levels, e.g. target values can be assigned at supplier’s perimeter level for supplier’s interfaces.
- Target values can be assigned to microcontrollers.
SPFM and LFM

Single Point Faults Metric:
- robustness of the item to single point and residual faults either by coverage from safety mechanisms or by design (primarily safe faults)

\[
\text{Single Point Fault metric} = 1 - \frac{\sum (\lambda_{\text{SPF}} + \lambda_{\text{RF}})}{\sum \lambda} = \frac{\sum (\lambda_{\text{MPF}} + \lambda_{\text{S}})}{\sum \lambda}
\]

Latent Faults Metric:
- robustness of the item to latent faults either by coverage of faults in safety mechanisms or by the driver recognizing that the fault exists before the violation of the safety goal, or by design (primarily safe faults)

\[
\text{Latent Fault metric} = 1 - \frac{\sum (\lambda_{\text{MPF,Latent}})}{\sum (\lambda - \lambda_{\text{SPF}} - \lambda_{\text{RF}})} = \frac{\sum (\lambda_{\text{MPF,perceived or detected}} + \lambda_{\text{S}})}{\sum (\lambda - \lambda_{\text{SPF}} - \lambda_{\text{RF}})}
\]
### Targets for relative metrics

<table>
<thead>
<tr>
<th></th>
<th>ASIL B</th>
<th>ASIL C</th>
<th>ASIL D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single point faults metric</strong></td>
<td>≥ 90 %</td>
<td>≥ 97 %</td>
<td>≥ 99 %</td>
</tr>
<tr>
<td><strong>Latent faults metric</strong></td>
<td>≥ 60 %</td>
<td>≥ 80 %</td>
<td>≥ 90 %</td>
</tr>
</tbody>
</table>

- In principle, target values could be derived from calculation applied to similar “well-trusted” design principles
  - well-trusted design = previously used without known safety anomalies
  - If a well-trusted design was assigned ASILD but had a SPFM=98% and the design is similar to the new one, the new design can reach ASILD with SPFM=98%
Aim of absolute metrics

To make available criteria that can be used in a rationale that the residual risk of a safety goal violation, due to random hardware failures of the item, is sufficiently low

- ‘Sufficiently low’ means “comparable to residual risks on items already in use”.

- The most common method consists of using a probabilistic metric called Probabilistic Metric for random Hardware Failures (PMHF)
The PMHF method

Computation of $\lambda_s$

Computation of DC / failure mode coverages

$\lambda_s$, $\lambda_{spf}$, $\lambda_{rf}$, $\lambda_{mpfd}$, $\lambda_{mpfp}$, $\lambda_{mpfl}$

PMHF
## Targets for absolute metrics

<table>
<thead>
<tr>
<th>ASIL</th>
<th>Random hardware failure target values</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>$&lt; 10^{-8} , h^{-1}$ (10 FIT)</td>
</tr>
<tr>
<td>C</td>
<td>$&lt; 10^{-7} , h^{-1}$ (100 FIT)</td>
</tr>
<tr>
<td>B</td>
<td>$&lt; 10^{-7} , h^{-1}$ (100 FIT)</td>
</tr>
</tbody>
</table>

- In principle, target values could be derived from calculation applied to similar “well-trusted” design principles
  - well-trusted design = previously used without known safety anomalies
  - If a well-trusted design was assigned ASILD but had a PMHF=15 FIT and the design is similar to the new one, the new design can reach ASILD with PMHF=15 FIT
HW Random Failures
HW random failures in a nutshell

Failure modes of a HW element

- Non-safety-related HW element
  - Safe fault
  - Safety-related HW element
  - Detected multiple point fault
  - Perceived multiple point fault
  - Latent multiple point Fault
  - Residual fault / single point fault

- Safe fault
- ASIL-D less than 10%
- ASIL-D less than 1%

Multiple point fault = at least the combination of a fault in the mission logic and a fault in the corresponding safety mechanism
HW random failures in a nutshell /2
FuSa architectural and absolute metrics

Single Point Fault metric = \[
\frac{\sum (\lambda_{MPF} + \lambda_S)}{\sum \lambda_{\text{Safety related HW elements}}}
\]

Latent Fault metric = \[
\frac{\sum (\lambda_{MPF \text{ perceived or detected}} + \lambda_S)}{\sum (\lambda - \lambda_{SPF} - \lambda_{RF})_{\text{Safety related HW elements}}}
\]

<table>
<thead>
<tr>
<th>ASIL</th>
<th>PMHF</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>&lt; 10^{-8} h^{-1} (10 FIT)</td>
</tr>
<tr>
<td>C</td>
<td>&lt; 10^{-7} h^{-1} (100 FIT)</td>
</tr>
<tr>
<td>B</td>
<td>&lt; 10^{-7} h^{-1} (100 FIT)</td>
</tr>
</tbody>
</table>

\[
PMHF = \lambda_{SPF} + \lambda_{RF} +
\]

\[
\lambda_{IF, DPF} \times 0.5 \times (\lambda_{SM, DPF, Latent} \times T_{\text{lifetime}}) +
\]

\[
\lambda_{IF, DPF} \times 0.5 \times (\lambda_{SM, DPF, Detected} \times T_{\text{service}}) +
\]

\[
\lambda_{SM, DPF} \times 0.5 \times (\lambda_{IF, DPF, Latent} \times T_{\text{lifetime}}) +
\]

\[
\lambda_{SM, DPF} \times 0.5 \times (\lambda_{IF, DPF, Latent} \times T_{\text{service}})
\]
Challenges to mitigate random failures

\[ \lambda_{RF} = \lambda \times \sum_{FM} \Lambda_{FM} \times (1 - F_{safeFM}) \times (1 - K_{FMC,RF_{FM}}) \]

- Failure rate of the safety related faults
- Failure modes distribution
- Fraction of safe faults
- Diagnostic Coverage: Fraction of residual faults prevented by safety mechanisms from violating the safety goal
Challenges to mitigate failure rates

\[ \lambda_{RF} = \lambda \times \sum_{FM} \Lambda_{FM} \times (1 - F_{safeFM}) \times (1 - K_{FMC,RF_{FM}}) \]

Failure rate of the safety related faults
Challenges to mitigate failure rates - hard faults

Related to the type of part

Related to temperature

Related to the die area and technology type

Related to package

Related to overstress

MATHEMATICAL MODEL:

\[ \lambda = \left( \lambda_1 \times N \times e^{-0.35 \times c} + \lambda_2 \right) \times \left( \frac{\sum_{i=1}^{y} \left( \pi_i \right) \times \tau_i}{\tau_{on} + \tau_{off}} \right) + \left( 2.75 \times 10^{-3} \times \pi_{on} \times \left( \sum_{i=1}^{\infty} \left( \frac{\pi_i}{\Delta T_i} \right)^{0.68} \right) \times \lambda_3 \right) + \left( \frac{\pi_1 \times \lambda_{EOS}}{\lambda_{overstress}} \right) \times 10^{-9} \times t \]

Source: IEC 62380
Challenges to mitigate failure rates – radiation effects

**Random**

- **HW**
  - **Permanent**
  - **Transient**

**Hard errors**

- *SEL = single event latch-up*

**Soft errors**

- *SEU = single event upset*
- *MBU = multiple bit upset*
- *SET = single event transient*

**Example:**

- *typical Soft Error Rate for a SRAM = 1000 FIT/Mbit*
- *If we consider 128KBytes RAM, the total raw failure rate is equal to λ = 1000 FIT !!!
Challenges to identify failure modes

\[ \lambda_{RF} = \lambda \times \sum_{FM} \Lambda_{FM} \times (1 - F_{safeFM}) \times (1 - K_{FMC,RFFM}) \]

Failure modes distribution
Challenges to identify failure modes - ISO 26262 view

Faults need to be identified and connect them to error and failures
Challenges to identify failure modes - examples

<table>
<thead>
<tr>
<th>FM Name</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coprocessor Pipeline/datapath/regbank leading to wrong data computation</td>
<td>39.18%</td>
</tr>
<tr>
<td>Breakpoints leading to wrong program flow execution</td>
<td>10.37%</td>
</tr>
<tr>
<td>Instruction cache leading to wrong program flow execution</td>
<td>4.77%</td>
</tr>
<tr>
<td>Load/Store and MMU control leading to wrong program flow execution</td>
<td>4.60%</td>
</tr>
<tr>
<td>Bus Interface Unit leading to wrong data management</td>
<td>3.55%</td>
</tr>
<tr>
<td>Main Register Bank leading to wrong data computation</td>
<td>3.30%</td>
</tr>
<tr>
<td>Load/store queue &amp; control - Watchpoints leading to wrong data computation</td>
<td>2.90%</td>
</tr>
<tr>
<td>Decoder unit (Dual) leading to wrong program execution</td>
<td>2.82%</td>
</tr>
<tr>
<td>Issue stage leading to wrong data computation</td>
<td>2.52%</td>
</tr>
<tr>
<td>Rename stage leading to wrong data computation</td>
<td>1.96%</td>
</tr>
<tr>
<td>MMU Buffers leading to wrong program flow execution</td>
<td>1.93%</td>
</tr>
<tr>
<td>Prefetch pipe and prediction logic leading to wrong program flow execution</td>
<td>1.89%</td>
</tr>
<tr>
<td>Store Buffer leading to wrong data management</td>
<td>1.65%</td>
</tr>
<tr>
<td>Instruction queue leading to wrong program flow execution</td>
<td>1.43%</td>
</tr>
<tr>
<td>Branch monitor and FIFO leading to wrong program flow execution</td>
<td>1.34%</td>
</tr>
<tr>
<td>MBIST interface leading to wrong program flow execution</td>
<td>1.26%</td>
</tr>
<tr>
<td>Dynamic Prediction queue leading to wrong program flow execution</td>
<td>1.23%</td>
</tr>
<tr>
<td>Execution unit (integer) 1/2 leading to wrong data computation</td>
<td>1.19%</td>
</tr>
<tr>
<td>MAC unit leading to wrong data computation</td>
<td>1.19%</td>
</tr>
</tbody>
</table>

Faults are NOT equi-distributed in failure modes!
Challenges to measure safe faults

\[ \lambda_{RF} = \lambda \times \sum_{FM} \Lambda_{FM} \times (1 - F_{safe_{FM}}) \times (1 - K_{FM\text{C},RF_{FM}}) \]

Fraction of safe faults
Challenges to measure safe faults - examples

\[ F_{SAFE_{tran}}^{FM} = F_{SAFE_{tran,arch}}^{FM} \cup F_{SAFE_{tran,sys}}^{FM} \]

**Brake by Wire**

- **Single bit-flips injected into CPU registers**
  - 40%: No impact
  - 33%: Detected by HW exception
  - 24%: Silent Data Corruption
  - 3%: Systems Hangs

**ABS**

- **43.3%**: Injected errors
- **30.7%**: Benign failure
- **24.3%**: Critical failure
- **1.4%**: Not recovered
Challenges to measure safe faults - vulnerability factors

AVF = the probability that a fault in a processor structure will result in a visible error in the final output of a program (introduced mainly by Shubhendu Sekhar Mukherjee, INTEL)

Table 2. AVF breakdown using Little’s Law. \#ACE inst = ACE IPC X ACE Latency. AVF = \#ACE inst / # instruction queue entries.

<table>
<thead>
<tr>
<th>Integer Benchmarks</th>
<th>ACE IPC</th>
<th>ACE Latency (cycles)</th>
<th># ACE Inst</th>
<th>AVF</th>
<th>Floating Point Benchmarks</th>
<th>ACE IPC</th>
<th>ACE Latency (cycles)</th>
<th>#ACE Inst</th>
<th>AVF</th>
</tr>
</thead>
<tbody>
<tr>
<td>bzip2-source</td>
<td>0.55</td>
<td>22</td>
<td>12</td>
<td>15%</td>
<td>ammp</td>
<td>0.23</td>
<td>92</td>
<td>21</td>
<td>33%</td>
</tr>
<tr>
<td>cc-200</td>
<td>0.57</td>
<td>18</td>
<td>10</td>
<td>16%</td>
<td>apphu</td>
<td>0.82</td>
<td>21</td>
<td>18</td>
<td>27%</td>
</tr>
<tr>
<td>crafty</td>
<td>0.37</td>
<td>15</td>
<td>6</td>
<td>9%</td>
<td>apsi</td>
<td>0.31</td>
<td>31</td>
<td>9</td>
<td>15%</td>
</tr>
<tr>
<td>eon-kajiya</td>
<td>0.36</td>
<td>20</td>
<td>7</td>
<td>11%</td>
<td>art-110</td>
<td>0.68</td>
<td>37</td>
<td>25</td>
<td>40%</td>
</tr>
<tr>
<td>gap</td>
<td>0.78</td>
<td>17</td>
<td>13</td>
<td>21%</td>
<td>equake</td>
<td>0.26</td>
<td>12</td>
<td>3</td>
<td>5%</td>
</tr>
<tr>
<td>gzip-graphic</td>
<td>0.60</td>
<td>13</td>
<td>8</td>
<td>12%</td>
<td>facerec</td>
<td>0.41</td>
<td>7</td>
<td>3</td>
<td>5%</td>
</tr>
<tr>
<td>mcf</td>
<td>0.25</td>
<td>68</td>
<td>17</td>
<td>26%</td>
<td>fma3d</td>
<td>0.59</td>
<td>11</td>
<td>7</td>
<td>10%</td>
</tr>
<tr>
<td>parser</td>
<td>0.49</td>
<td>24</td>
<td>12</td>
<td>15%</td>
<td>galgel</td>
<td>1.10</td>
<td>21</td>
<td>23</td>
<td>35%</td>
</tr>
<tr>
<td>perlbench-makerand</td>
<td>0.38</td>
<td>17</td>
<td>7</td>
<td>10%</td>
<td>lucas</td>
<td>1.23</td>
<td>17</td>
<td>21</td>
<td>33%</td>
</tr>
<tr>
<td>twolf</td>
<td>0.30</td>
<td>27</td>
<td>8</td>
<td>13%</td>
<td>mesa</td>
<td>0.47</td>
<td>16</td>
<td>8</td>
<td>12%</td>
</tr>
<tr>
<td>vortex lendiant3</td>
<td>0.42</td>
<td>22</td>
<td>9</td>
<td>15%</td>
<td>mgrid</td>
<td>1.28</td>
<td>10</td>
<td>13</td>
<td>21%</td>
</tr>
<tr>
<td>vpr-route</td>
<td>0.35</td>
<td>12</td>
<td>4</td>
<td>7%</td>
<td>sixtrack</td>
<td>0.66</td>
<td>20</td>
<td>13</td>
<td>21%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>swim</td>
<td>1.08</td>
<td>16</td>
<td>17</td>
<td>27%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wnpwise</td>
<td>1.60</td>
<td>13</td>
<td>20</td>
<td>31%</td>
</tr>
<tr>
<td>average</td>
<td>0.45</td>
<td>23</td>
<td>9</td>
<td>15%</td>
<td>average</td>
<td>0.77</td>
<td>23</td>
<td>14</td>
<td>23%</td>
</tr>
</tbody>
</table>

36th Annual International Symposium on Microarchitecture (MICRO), December 2003
Challenges to measure safe faults – AVF example

<table>
<thead>
<tr>
<th>FM Name</th>
<th>Fsafe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coprocessor pipeline/datapath/regbank leading to wrong data computation</td>
<td>99.80%</td>
</tr>
<tr>
<td>Breakpoints leading to wrong program flow execution</td>
<td>49.32%</td>
</tr>
<tr>
<td>Instruction cache leading to wrong program flow execution</td>
<td>48.37%</td>
</tr>
<tr>
<td>Load/Store and MMU control leading to wrong program flow execution</td>
<td>49.32%</td>
</tr>
<tr>
<td>Bus Interface Unit leading to wrong data management</td>
<td>49.32%</td>
</tr>
<tr>
<td>Main Register Bank leading to wrong data computation</td>
<td>49.32%</td>
</tr>
<tr>
<td>Load/store queue &amp; control - Watchpoints leading to wrong data computation</td>
<td>49.32%</td>
</tr>
<tr>
<td>Decoder unit (Dual) leading to wrong program execution</td>
<td>48.82%</td>
</tr>
<tr>
<td>Issue stage leading to wrong data computation</td>
<td>49.32%</td>
</tr>
<tr>
<td>Rename stage leading to wrong data computation</td>
<td>49.32%</td>
</tr>
<tr>
<td>MMU Buffers leading to wrong program flow execution</td>
<td>49.32%</td>
</tr>
<tr>
<td>Prefetch pipe and prediction logic leading to wrong program flow execution</td>
<td>48.37%</td>
</tr>
<tr>
<td>Store Buffer leading to wrong data management</td>
<td>49.32%</td>
</tr>
<tr>
<td>Instruction queue leading to wrong program flow execution</td>
<td>48.37%</td>
</tr>
<tr>
<td>Branch monitor and FIFO leading to wrong program flow execution</td>
<td>72.52%</td>
</tr>
<tr>
<td>MBIST interface leading to wrong program flow execution</td>
<td>99.50%</td>
</tr>
<tr>
<td>Dynamic Prediction queue leading to wrong program flow execution</td>
<td>48.37%</td>
</tr>
<tr>
<td>Execution unit (integer) 1/2 leading to wrong data computation</td>
<td>82.48%</td>
</tr>
<tr>
<td>MAC unit leading to wrong data computation</td>
<td>99.97%</td>
</tr>
</tbody>
</table>

Faults are NOT all dangerous!
Challenges to measure safe faults

IC supplier → End users (Tiers and OEM)

- Soft error
- Wrong line
- Wrong object
- Unintentional braking event

fault models → failure mode at image level → failure mode at algorithm level → hazard at item level
Challenges to detect random failures

\[ \lambda_{RF} = \lambda \times \sum_{FM} \Lambda_{FM} \times (1 - F_{safeFM}) \times (1 - K_{FMC,RF_{FM}}) \]

Diagnostic Coverage

Fraction of residual faults prevented by safety mechanisms from violating the safety goal
Challenges to detect random failures - options

- **SW/system measures** implemented by end user with or without reference implementation
  - FuSa-specific SW Test Libraries
  - FuSa-specific HW measures
  - «Intrinsic» HW measures (e.g. ECC)
Challenges to detect random failures – ECC pitfalls

How MBU can impact ECC diagnostic coverage:

\[ k_{\text{RAM}}^{\text{tran}} = \frac{k_{\text{SEC/DED tran}}^{\text{RAM}} \times \lambda_{\text{RAM}} \cdot (1 - \text{MBU} \%) + k_{\text{MBU tran}}^{\text{RAM}} \times \lambda_{\text{RAM}} \cdot \text{MBU} \%}{\lambda_{\text{RAM}}} \]

• If we assume:
  \[ k_{\text{SEC/DED tran}}^{\text{RAM}} = 100\% \]
  \[ k_{\text{MBU tran}}^{\text{RAM}} = 68\% \]

<table>
<thead>
<tr>
<th>MBU%</th>
<th>[ k_{\text{RAM}}^{\text{tran}} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>96,8%</td>
</tr>
<tr>
<td>1%</td>
<td>99,680%</td>
</tr>
<tr>
<td>0,1%</td>
<td>99,9680%</td>
</tr>
<tr>
<td>0,01%</td>
<td>99,99680%</td>
</tr>
</tbody>
</table>
Challenges to detect random failures – in field tests

• **Availability of Memory and Logic in field tests is key:**
  - Memory test = it shall be executed at power-on and at run-time on all safety critical memories and arrays, it shall cover different failure modes (including AF, ADF), it shall be quick (less than few ms)
  - Logic test = it shall be executed at least at power-on and possibly at run-time, at least on cores. It shall cover at least stuck-at faults.
  - Typical target coverage is at least 90%.

• **Fault forecasting/prediction is an important feature** (ref. ISO 26262-11)
Challenges to detect random failures - SW Tests

- SW Test Library (STL) executed periodically (e.g. each 100ms) to test a specific core
  - For permanent faults (ASILB, 90%)
- The diagnostic coverage of the STL shall be verified by means of fault injection at gate level
- STL shall be developed according to safety standards

Each TS:
- targets a specific function or a group of functions of the component
- provides pass/fail information and self-checking signatures (CRC)
- may be interrupted at any time by the application SW
Challenges to detect random failures - “safe island”

«Safe island» or «safe MCU on chip», i.e. an/ASILD subsystem inside the SoC, typically implemented with two medium performance cores in HW lock-step, responsible of «FuSa housekeeping», interfacing with in vehicle networks and running SW to test the other portion of the SoC

- **Smart Comparison**
  - Fault discrimination
  - Functional downgrade
  - Latent faults detection
  - Timeout on system reset exiting
  - Common Cause Failures detection
Challenges to detect random failures – LCLS

SW tasks fully replicated and running on two cores (global redundancy). Additional SW making the comparison task (usually done in an independent core).

Main ISO 26262 requirements
- Avoidance of common cause failures
- Verification of the diagnostic coverage achieved as function of type/frequency of checks and type of algorithm

It seems an easy concept but it has many pitfalls (e.g. RTOS)
Challenges to detect random failures – other redundancies

Proposed by the Association of the German Automotive Industry (VDA) as standardized monitoring concept for the engine control of gasoline and diesel engines.

The max ASIL achievable by VDA E-Gas is typically limited to ASILC.

The coverage is highly application dependent – i.e. Level1/Level2 and even Level 3 shall be rewritten for each specific application.
Fault tolerant systems in automotive - examples /1
Fault tolerant systems in automotive - examples /2
Conclusion