June 25th, 2007

Functional Safety and Safety Standards: Challenges and Comparison of Solutions

Christopher Temple
Automotive Systems Technology Manager
Overview

► Functional Safety Basics
► Functional Safety Standards
► Functional Safety Measures
Functional Safety Basics
Functional Safety Definition

Definitions according to IEC61508

► “risk”
  • “combination of the probability of occurrence of harm and the severity of that harm”

► “harm”
  • “physical injury or damage to the health of people either directly or indirectly as a result of damage to property or to the environment”

► “safety”
  • “freedom from unacceptable risk”

► “functional safety”
  • “part of the overall safety relating to the equipment under control (EUC) and the EUC control system which depends on the correct functioning of the electrical/electronic/programmable electronic (E/E/PE) safety-related systems, other technology related safety-related systems and external risk reduction facilities”
Safety in the Context of Dependability

Definitions according to IFIP WG10.4

- **Reliability**: Probability that the system will perform to its specification throughout a period of duration $t$.
- **Availability**: Percentage of time for which the system will perform to its specification.
- **Safety**: Probability that the system will *not show a specified dangerous behavior* throughout a period of duration $t$.
- **Dependability**: Secondary attributes: *Usability, recoverability, maintainability, extendibility, trustability, etc.*

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2007-2008.
## Impairments through Faults

<table>
<thead>
<tr>
<th>Phenom. cause</th>
<th>Phase of creation</th>
<th>System boundary</th>
<th>Persistence</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design faults</td>
<td>man-made</td>
<td>development</td>
<td>internal (HW/SW)</td>
<td>permanent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>architecture, algorithms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>temporary</td>
</tr>
<tr>
<td>Manufacturing faults</td>
<td>man-made</td>
<td>manufacturing</td>
<td>internal (HW/SW)</td>
<td>permanent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>compiler bugs, test escapes, SI-faults</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>temporary</td>
</tr>
<tr>
<td>Physical faults</td>
<td>physical</td>
<td>operational</td>
<td>internal</td>
<td>permanent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ageing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>intermittent</td>
<td>timing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>external</td>
<td>permanent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>noise, short circuit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transient</td>
<td>a-particles, glitches</td>
</tr>
<tr>
<td>Interaction faults</td>
<td>man-made</td>
<td>operational</td>
<td>external</td>
<td>temporary</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>interface formats</td>
</tr>
</tbody>
</table>
Fault - Error - Failure Model

Fault

error latency

Error
detected

Failure

error activation
detected

cause of an error

fault dormancy

fault duration

fault occurrence

error latency

fault activation
Fault - Error - Failure Model

Fault

Error

Failure

**Error**
Manifestation of the fault in a system

Fault occurrence

fault duration

Fault dormancy

error latency

error detection latency

error detection

error activation

error activation
Fault - Error - Failure Model

Fault

Error latency

Error

Fault dormancy

Failure

Deviation of the delivered service from compliance with the specification
A system can be viewed as a set of interacting components with each component being a system in itself but on a lower hierarchy level.
System / Component Hierarchy

functional safety develops top down

System (top hierarchy)

Component  Component  Component

Component is a new system in itself

Component  Component  Component

“safety measures” work towards functional safety objective bottom up
# Addressing Faults in Systems

## Means for Addressing Faults in Systems

<table>
<thead>
<tr>
<th>Means for Addressing Faults</th>
<th>Development Lifecycle</th>
<th>HW Measures</th>
<th>SW Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Prevention</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fault Tolerance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(Error Processing, Fault Treatment)</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fault Removal</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(Verification, Diagnosis, Correction)</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fault Forecasting</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Qualitative Evaluation, Quantitative Evaluation)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Functional Safety Standards
Safety Challenge

► No established metric and value network to classify safety of measures without top system context
  • How to quantify safety of measures?
  • How to compare different approaches?
  • How to make a good techno-economical decision?
Role of Standards

Standards are emerging as a framework to establish metrics
- IEC61508 (existing)
  - Safety lifecycle defined
  - Top down
  - Recommended & mandatory practices
- ISO26262 (emerging)
  - Decomposition of safety from system to component level
The 7 Parts of IEC 61508

► 1: General Requirements
► 2: Requirements for electrical / electronic / programmable electronic safety-related systems (means HW)
► 3: Software Requirements
► 4: Definitions and abbreviations
► 5: Examples of methods for the determination of safety integrity levels
► 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3
► 7: Overview of techniques and measures
IEC61508 Safety Lifecycle

Concept

Overall scope definition

Hazard & risk analysis

Overall safety requirements

Safety requirements allocation

Outside of the scope of IEC61508

Overall planning

Overall operation and maintenance planning

Overall safety validation planning

Overall installation and commissioning planning

Safety-related systems: E/E/PES

Realization

Safety-related systems: other technology

Realization

External risk reduction facilities

Realization

Overall installation and commissioning

Overall safety validation

Overall operation, maintenance and repair

Decommissioning or disposal

Back to appropriate overall safety lifecycle phase

Overall modification and retrofit
Carry out a failure mode and effect analysis to determine the effect of each failure mode of each component or group of components in the subsystem on the behaviour of the E/E/PE safety-related systems.

Following is required:

- a detailed block diagram of the E/E/PE safety-related system describing the subsystem together with the interconnections for that part of the E/E/PE safety-related system which will affect the safety function(s) under consideration;

- the hardware schematics of the subsystem describing each component or group of components and the interconnections between components;

- the failure modes and rates of each component or group of components and associated percentages of the total failure probability corresponding to safe and dangerous failures.
Interaction between IEC61508 Part 2 and Part 3

Source: IEC61508-3, "7 Software safety lifecycle requirements", figure 7
### Target Failure Rates According To IEC61508

**Table 3 – Safety integrity levels: target failure measures for a safety function operating in high demand or continuous mode of operation**

<table>
<thead>
<tr>
<th>Safety integrity level</th>
<th>High demand or continuous mode of operation (Probability of a dangerous failure per hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$\geq 10^{-9} \text{ to } &lt; 10^{-8}$</td>
</tr>
<tr>
<td>3</td>
<td>$\geq 10^{-8} \text{ to } &lt; 10^{-7}$</td>
</tr>
<tr>
<td>2</td>
<td>$\geq 10^{-7} \text{ to } &lt; 10^{-6}$</td>
</tr>
<tr>
<td>1</td>
<td>$\geq 10^{-6} \text{ to } &lt; 10^{-5}$</td>
</tr>
</tbody>
</table>

**NOTE** – See notes 3 to 9 below for details on interpreting this table.

Diagnostic Coverage & Safe Failure Fraction

Current definition in IEC61508

```
„diagnostic coverage“  DC   =   \( \frac{\sum \lambda_{dd}}{\sum \lambda_d} \)
„safe failure fraction“ SFF  =  \( \frac{\sum \lambda_s + \sum \lambda_{dd}}{\sum \lambda_s + \sum \lambda_d} \)  

= \( \frac{\sum \lambda_s + DC \times \sum \lambda_d}{\sum \lambda_s + \sum \lambda_d} \)

with \( \lambda_s = 0 \):

SFF  =  DC

\( \sum \lambda_s \):    safe failure probability  
\( \sum \lambda_d \):    dangerous failure probability

\( \sum \lambda_{dd} \):    detected dangerous failure probability  
\( \sum \lambda_{ud} \):    undetected dangerous failure probability

\( \sum \lambda_d \) = \( \sum \lambda_{dd} + \sum \lambda_{ud} \)
```
Designing a Safe System

Hazard Analysis
Which unintended situations (hazards) can occur?

Risk Analysis
How likely is a hazard? How dangerous is a hazard? How controllable is the system in case of a hazard?

Safety Integrity Level 1..4
Dangerous failure rate $\lambda_{du}$
Diagnostic Coverage DC
Safe Failure Fraction SFF

Safety Functions
How to mitigate the hazards?

Integrity Requirements
Are the safety functions effective enough?

Refine the system until the remaining risk is below the highest acceptable risk
### IEC 61508 versus ISO 26262

<table>
<thead>
<tr>
<th></th>
<th>IEC 61508</th>
<th>ISO 26262</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target</strong></td>
<td>General safety standard for E/E/PE systems</td>
<td>Adaption of IEC 61508 for the automotive industry</td>
</tr>
<tr>
<td><strong>Risk classification</strong></td>
<td>SIL1 - SIL4</td>
<td>ASIL A - ASIL D</td>
</tr>
<tr>
<td><strong>Development Lifecycle</strong></td>
<td>Overall safety lifecycle</td>
<td>Automotive safety lifecycle (incl. V model)</td>
</tr>
<tr>
<td><strong>Recommended &amp; mandatory HW measures &amp; practices</strong></td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td><strong>Recommended &amp; mandatory SW measures &amp; practices</strong></td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td>Safety Integrity Level</td>
<td>SIL 1</td>
<td>SIL 2</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>IEC61508 [today]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ranking by assessing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the probability of a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dangerous failure per</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASIL A</td>
<td>ASIL B</td>
<td>ASIL C</td>
</tr>
<tr>
<td>ISO26262 [future]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ranking by assessing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>severity of injuries,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exposure to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hazardous situations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>and the controllability of the driving situation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Direct comparison not possible
- Assessment of recommended & mandatory HW/SW measures & practices
Functional Safety Measures
## Impairments through Faults

<table>
<thead>
<tr>
<th>Phenom. cause</th>
<th>Phase of creation</th>
<th>System boundary</th>
<th>Persistence</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design faults</td>
<td>man-made</td>
<td>development</td>
<td>internal (HW/SW)</td>
<td>permanent temporary architecture, algorithms</td>
</tr>
<tr>
<td>Manufact. faults</td>
<td>man-made</td>
<td>manufacturing</td>
<td>internal (HW/SW)</td>
<td>permanent temporary compiler bugs, test escapes, SI-faults</td>
</tr>
<tr>
<td>Physical faults</td>
<td>physical</td>
<td>operational</td>
<td>internal</td>
<td>permanent ageing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>intermittent</td>
<td>timing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>external</td>
<td>permanent noise, short circuit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transient</td>
<td>a-particles, glitches</td>
</tr>
<tr>
<td>Interaction faults</td>
<td>man-made</td>
<td>operational</td>
<td>external</td>
<td>temporary interface formats</td>
</tr>
</tbody>
</table>
### Addressing Faults in Systems

<table>
<thead>
<tr>
<th>Means for Addressing Faults</th>
<th>Development Lifecycle</th>
<th>HW Measures</th>
<th>SW Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Prevention</td>
<td>✅</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault Tolerance (Error Processing, Fault Treatment)</td>
<td></td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Fault Removal (Verification, Diagnosis, Correction)</td>
<td></td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Fault Forecasting (Qualitative Evaluation, Quantitative Evaluation)</td>
<td></td>
<td></td>
<td>☑</td>
</tr>
</tbody>
</table>

TM Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2007-2008.
## Error Processing versus Fault Treatment

<table>
<thead>
<tr>
<th>Error Processing</th>
<th>Fault Treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error Detection</strong>&lt;br&gt; (detect latent errors to mitigate effect of error)</td>
<td><strong>Error Diagnostics</strong>&lt;br&gt; ‘forces’ activation of fault within test interval</td>
</tr>
<tr>
<td>Detects error upon activation by ‘normal’ system operation</td>
<td><strong>Error Recovery</strong>&lt;br&gt; Backward &amp; forward recovery</td>
</tr>
<tr>
<td>Duplexing &amp; Comparison, …&lt;br&gt; (Redundancy based techniques)&lt;br&gt; Timing &amp; Execution Checks&lt;br&gt; (Reasonableness based techniques)</td>
<td><strong>Compensation</strong>&lt;br&gt; (Redundancy based techniques)</td>
</tr>
<tr>
<td><strong>Testing &amp; Fault Diagnosis</strong>&lt;br&gt; (detect dormant faults before these create errors)</td>
<td><strong>Online BIST</strong>&lt;br&gt; (Redundancy based techniques, Reasonableness based techniques)</td>
</tr>
<tr>
<td><strong>Fault Isolation</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Reconfiguration</strong></td>
<td></td>
</tr>
</tbody>
</table>

In general error detection ≠ online testing!
Tradeoffs of Different Redundancy Approaches

Time related approaches

- different clock cycles
- concurrent threads
- one after another

⇒ Tradeoff: Performance

Hardware related approaches

- different chips
- different die areas
- different modules
- different submodules
- different FFs

⇒ Tradeoff: HW Costs

Algorithm related approaches

- plausibility check
- main prg & math/flow
- float & integer
- different check & response
- different math/flow

⇒ Tradeoff: SW Complexity
Processing Subsystem Philosophies for Safety

**Master / Slave Approach**

**Dual Processor Approach**

**Single Core Self Test Approach**

**Dual Core Approach**
Comparison of different Architectures

Key Parameters

► Desirable System Properties
  • Low Cost
  • Low Complexity
  • High Availability

► Safety Properties
  • Transient Fault Detection
  • Early Detection of permanent Faults
  • Detection of systematic SW Faults

Architectures

<table>
<thead>
<tr>
<th>#Cores</th>
<th>coupling</th>
<th>self test</th>
</tr>
</thead>
<tbody>
<tr>
<td>single core</td>
<td>n.a.</td>
<td>no</td>
</tr>
<tr>
<td>dual core</td>
<td>locked</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>&quot;tightly coupled&quot;</td>
<td>yes</td>
</tr>
<tr>
<td>dual core</td>
<td>&quot;loosely coupled&quot;</td>
<td>yes</td>
</tr>
<tr>
<td>multi-core</td>
<td>n.a.</td>
<td></td>
</tr>
<tr>
<td>multi-device</td>
<td>n.a.</td>
<td></td>
</tr>
</tbody>
</table>
Metric for Fault Tolerance Mechanisms

Desirable System Properties

Fault Detection

Detection of transient Faults

Detection of systematic SW Faults

Detection of permanent Faults

Availability

Low Complexity

Low Cost
Dual Core - Fault Tolerance Mechanisms

Low Complexity

Availability

Detection of transient Faults

Detection of systematic SW Faults

Low Cost

dual core locked no self test

dual core "tightly coupled" with self test

dual core locked with self test

<table>
<thead>
<tr>
<th>#Cores</th>
<th>coupling</th>
<th>self test</th>
</tr>
</thead>
<tbody>
<tr>
<td>single core</td>
<td>n.a.</td>
<td>yes</td>
</tr>
<tr>
<td>dual core</td>
<td>locked</td>
<td>yes</td>
</tr>
<tr>
<td>dual core</td>
<td>&quot;tightly coupled&quot;</td>
<td>yes</td>
</tr>
<tr>
<td>multi-core</td>
<td>n.a.</td>
<td></td>
</tr>
<tr>
<td>multi-device</td>
<td>n.a.</td>
<td></td>
</tr>
</tbody>
</table>

early Detection of permanent Faults
“Multi” Core/Device - Fault Tolerance Mechanisms

- Low Complexity
- Detection of transient Faults
  - dual (identical) core “loosely coupled”
- Availability
- Detection of systematic SW Faults
  - core with coprocessor
- Low Cost
- early Detection of permanent Faults
  - dual device

<table>
<thead>
<tr>
<th>#Cores</th>
<th>coupling</th>
<th>self test</th>
</tr>
</thead>
<tbody>
<tr>
<td>single core</td>
<td>n.a.</td>
<td>no</td>
</tr>
<tr>
<td>dual core</td>
<td>locked</td>
<td>yes</td>
</tr>
<tr>
<td>dual core</td>
<td>“loosely coupled”</td>
<td>yes</td>
</tr>
<tr>
<td>multi-core</td>
<td>n.a.</td>
<td></td>
</tr>
<tr>
<td>multi-device</td>
<td>n.a.</td>
<td></td>
</tr>
</tbody>
</table>
Summary

► Functional Safety Basics
  • Functional safety is a system property
  • It is impaired by faults

► Functional Safety Standards
  • IEC61508 / ISO26262
  • Top down assessment
  • Safety lifecycle
  • Recommended & mandatory HW/SW measures & practices

► Functional Safety Measures
  • Error detection versus testing
  • Redundancy
  • Trade-off depending on fault assumptions