ARM

White Paper

# The Functional Safety Imperative in Automotive Design

Functional safety is about ensuring that products operate safely and do so even when they go wrong. Here's why you can't choose to ignore it.

Andrew Hopkins – Strategic Technology Specialist

September 2016

# Introduction

The automotive industry has entered a period of rapid evolution that is changing the way cars are designed, used and sold. Driver safety technologies, traffic congestion, environmental concern, and the fundamental premise of how we use our cars are all influencing the next generation of vehicles. Many OEMs are addressing these challenges through increased computation for greater control. As a sign of the times, Euro NCAP continues to push the vehicle manufacturers by regularly evolving the five star rating to include more and more safety assist features such as lane support. The number of processors in each market segment has progressively risen and there is now an average of 40-50 processors per car with high-end models containing around 120 processors. Moreover, Semicast Research forecasts annual revenues for under-the-hood automotive ECU electronics alone will grow to almost USD 86 billion by 2022, from around USD 53 billion in 2015, a CAGR of seven percent making automotive electronics a valuable opportunity for semiconductor vendors.

These chips will improve powertrain emissions, enhance safety and provide connectivity to other cars and road infrastructure using cellular networks. However these sophisticated systems require a fool-proof way to keep drivers safe, which is called Functional Safety. Addressing the needs of functional safety is essential for success in the automotive industry, however it presents challenges to even the most experienced of semiconductor vendors.

# What's functional safety all about?

In a nut-shell, functional safety is about ensuring the safe operation of systems even when they go wrong; this sets the cultural mind-set behind ARM's market-agnostic developments relevant to functional safety.

Each industry typically has a standard to guide developments and set minimum expectations, and for automotive electronics it is ISO 26262, which defines functional safety as:

*The absence of unreasonable risk due to hazards caused by malfunctioning behaviour of electrical / electronic systems*

Standards for other markets, such as IEC 61508 for electrical and electronic systems and DO-254 for airborne electronic hardware, have their own definitions, although more importantly they also set their own terminology and guidance for engineering developments including target metrics. Hence it's important to identify the target markets before starting development and ensure suitable processes are followed, as attempts to retrofit development processes can be inefficient. Figure 1 illustrates a variety of standards applicable to silicon IP. In practice it is possible to address the needs of multiple standards by identifying their specific requirements and adopting common principles such as quality management and a focus on safety from the outset.

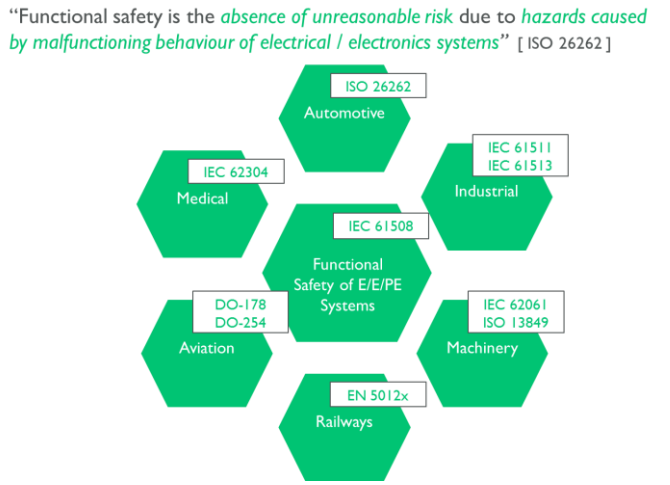


*Figure 1: Standards for functional safety of silicon IP*

In practice, functional safety means a system that is demonstrably safe to an independent assessor in accordance with the target standards. Safety requires predictable failure modes which could be with full functionality, graceful degradation in functionality or a clean shutdown followed by a reset and restart.

Not all faults will lead to hazardous events immediately. For example a fault in a car's power steering might lead

to incorrect sudden steering action. However, since the electronic and mechanical designs will have natural timing delays, faults can be tolerated for a specific amount of time, often many milliseconds or more. In the ISO 26262 this time is known as the fault tolerant time interval, and depends on the potential hazardous event and the system design. Naturally the more safety critical the application, the more unsafe faults must be mitigated.

In the ideal world functional safety would have no impact on system performance, although in practice many of the counter measures available to designers can significantly affect performance, power and area. So one of the challenges is to achieve sufficient functional safety whilst trying not to adversely affect the system or its cost to design or manufacture.

## Why do you need Functional Safety?

Functional safety for Silicon IP used to be a niche activity, limited to a small circle of chip and system developers in automotive, industrial, aerospace and similar markets. However over the last few years this has changed significantly as the variety of automotive applications has grown. Furthermore, there are many markets that will benefit hugely from the introduction of more electronics, so long as the systems are functionally safe. Medical electronics and aviation are two examples.

Autonomous driving has caught a lot of attention over the last few years, with a vast amount of tests done by technology and automotive companies alike. The future looks very exciting with a growing number of increasingly capable Advanced Driver Assistance Systems (ADAS) to capture people's interest and media-rich In-Vehicle Infotainment (IVI), but much work remains until highly autonomous driving comes of age. The emergence of drones in all shapes and sizes and the growing ubiquity of industrial Internet of Things are also interesting examples where functional safety is needed, and ARM® technology offers a significant advantage.

## ARM technology for functional safety

Much like any technology market gaining traction these burgeoning applications require semiconductors to make them happen and the fast-pace of product innovation has attracted huge interest from ARM's partners. One of the essential elements of most functionally safe embedded systems is a combination of a safety and real-time capability. To address these needs, the ARM Cortex-R family of processors offer high-performance computing solutions for embedded systems where reliability, high availability, fault tolerance and/or deterministic real-time responses are needed. These features provide a foundation for a high-integrity safety island within devices for ADAS and IVI where they can be used to take on the most critical of processing activities, service safety related interrupts, communicate with other systems and act as a supervisor for complicated capabilities at a lower integrity.

## What's a fault?

Failures can be systematic to the design itself, such as human errors in specifications and design, or due to the tools used. One way to reduce these errors is to have rigorous quality processes that include a range of plans, reviews and measured assessments. Being able to manage and track requirements is also important, as is good planning and qualification of the tools to be used. ARM provides ARM Compiler 5 certified by TÜV SÜD to enable safety-related development without further compiler qualification.

Another class of failure is random hardware faults; they could be permanent faults such as a short circuit between wires, pins or tracks as illustrated by Figure 2. Alternatively they could be soft errors caused by exposure to the natural radiation all around us. Such faults can be detected by counter measures designed into the hardware and software. System-level approaches are also important, for example Logic Built-In-Self-Test (BIST) can be applied at start up or shutdown in order to distinguish between soft and permanent faults.

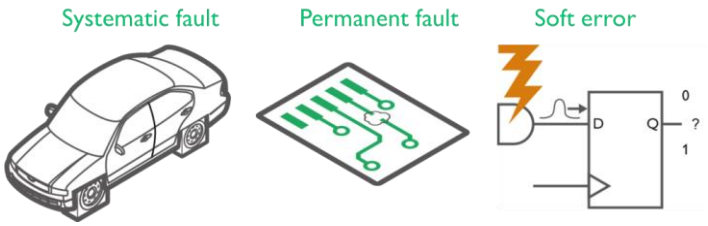


*Figure 2: Classes of fault*

## Countermeasures

Selection and design of countermeasures for fault detection and control is part of the process where designers can add a lot of value, as they get to apply both system-level and microarchitecture techniques to overcome a challenge. A good way to start is with a concept-level Failure Modes and Effects Analysis (FMEA) to identify the list of possible failure modes in the system and how significant their effects are. Armed with that list and an understanding of the system's complexity, the most significant failure modes can be identified and countermeasures designed.

There are many ways to address potential failures, and some of the common techniques are:

- Diverse checkers. Use of a different circuit to identify whether the main circuit has failed. An example checker is a scoreboard for an interrupt controller that keeps a tally of the interrupts requested and those raised to the system.
- Full lock-step replication. This technique is employed for Cortex-R5 and involves instancing a substantial IP component, such as a processor, more than once with operation delayed in time by a few cycles to create temporal and spatial redundancy. The larger memories are often shared between the instances to lower the area needed as although very robust, this technique is also expensive.
- Selective hardware redundancy. Here only a critical part of the hardware may be replicated, such as an arbiter.
- Software redundancy. The overhead and complexity of hardware redundancy is not always needed and is an inflexible use of resources. An alternative for processing is to compute a calculation more than once on different processor cores, and check the results match.
- Error detection and correcting codes are another well-known technique which is generally used to protect memory and buses. There are many different types of code, however the aim is to gain redundancy through a few additional bits rather than fully replicate the underlying data. The state-of-the-art for automotive systems is to employ sufficient redundancy to detect a two bit error in a memory word. Optionally correction can also be applied.

## Fault logging

Once faults are detected it is essential they are logged so that supervisor software can determine the health of the system and ensure it is still safe. Separation of safe faults such as memory corrections and unsafe faults such as irreparable hardware errors must be recorded separately.

Fault logging generally starts with counting faults, either by system-level infrastructure that counts signalled events, akin to interrupts, or by counters within the IP itself. In order to help make sense of these events it is also desirable to know what caused them, starting with the oldest events. To support this requirement and facilitate debugging, some IP may capture additional details such as the detected memory address which should normally persist through a soft reset so they can be read at start-up and any diagnostics like self-tests run.

It's important to remember that faults can occur in the safety infrastructure too. Unlike faults in the active

hardware that will typically be encountered quickly during use, faults in the safety checkers can remain dormant and allow unsafe faults to propagate undetected. Such faults are known as latent faults and can be mitigated by periodic testing of the safety checkers.

## Safety integrity levels

Within the standards there are different levels of safety defined to reflect the function's criticality. For example, ECUs controlling the windscreen wipers, airbags or brakes need to have a higher integrity than those controlling the speedometer or parking sensors because vision through the windscreen is essential and unintended braking or airbag deployment could be fatal. On the other hand, neither the speedometer nor parking sensors are essential to a human stopping a car safely. However, with the introduction of ADAS and highly autonomous driving where the electronics assume increased control of the vehicle the required integrity and therefore safety engineering are also rising.

The level of integrity needed is therefore linked to the necessity and ability of a human to avert an unsafe situation, and the standards provide guidance to help qualify the level of integrity needed and metrics to quantify the integrity of systems.

IEC 61508 defines a range of Safety Integrity Levels (SIL) ranging from 1 to 4 where 4 is the highest integrity. Likewise ISO 26262 defines a concept of Automotive SILs (ASIL) ranging from the lowest level, ASIL A, to the highest integrity of ASIL D. ISO 26262 proposes metrics for so called single point faults, latent faults and a Probabilistic Metric of Hardware Failure (PMHF) – known by the enterprise market as failure in time – for ASIL B to ASIL D as shown by Table 2. The proportion of detectable faults is known as diagnostic coverage.

| Proposed Target | ASIL B | ASIL C | ASIL D |
|---|---|---|---|
| Single point fault | ≥90% | ≥97% | ≥99% |
| Latent faults | ≥60% | ≥80% | ≥90% |
| PMHF | $<10^{-7}$ / h | $<10^{-7}$ / h | $<10^{-8}$ / h |

*Table 1. ISO 26262 proposed metrics*

These metrics are often seen as a normative requirement, although in practice they are a proposal for automotive applications, and developers can justify their own target metrics because the objective is to enable safe products, not add bullet points to a product datasheet. Going back to the examples given previously, windscreen wipers, braking and airbag deployment could be classified as ASIL D, whereas the speedometer and parking sensors might constitute ASIL B or lower depending on the overall system design and the safety case.

Irrespective of the diagnostic coverage achieved, it is essential to follow suitable processes when targeting functionally safe applications – and this is where the standards really help. Furthermore, rigorous quality processes can improve overall product quality for any application regardless of whether functional safety applies.

## Processes for functionally safe IP design

An essential part of developing IP for functionally safe applications is the process followed which must include consideration of safety from the outset and the establishment of a supportive safety culture.

The main aspects the processes must cover are:

- Safety management, which covers team organisation including clear definition of roles and responsibilities, safety culture, a definition of the safety life cycle and the definition of the functional safety support levels. Part of the safety lifecycle is a plan for success and to select suitable tools to support the development and ensure the teams have adequate training.
- Requirements management and traceability of fault detection and control features (countermeasures). In order to trace requirements they need to be clearly, uniquely and atomically defined. The level of traceability depends on the integrity requirements and can be high-level from document to document or in detail from product requirement through to the verification results for fault detection and control features – all of which must be thoroughly verified.
- Quality management goes beyond requirements traceability, errata must also be managed and handled appropriately for which ARM has many years of experience. Documenting and communicating the processes followed is also important.

## The safety documentation package

The development of IP is one way ARM supports partners; however the relationship does not end when the IP is delivered. For functional safety related IP developments ARM defines two levels of safety documentation package:

- Standard to support use up to ASIL B
- Extended for use up to ASIL D

Each safety documentation package comprises a safety manual that describes the process followed, the fault detection and control features and assumptions of use along with other details. There is also a Failure Modes and Effects Analysis report to provide an example of the diagnostic coverage achievable using the IP and support further analysis at chip-level by partners. The package also includes a clear definition of the development interface between ARM and the licensee.

## Safety Element out of Context

Key to functional safety is the preparation of a safety case. This is a structured argument, supported with evidence, which justifies why the system is acceptably safe in its specific operating environment. Safety cases are also usually hierarchical. The safety case composed by SoC developers is composed of input from each IP supplier which then enables their customer and so forth. Most licensable silicon IP will be developed as a so called Safety Element out of Context (SEooC), where its designers will have no specific understanding of how it will be subsequently be used. Hence the safety manual must also capture insight from the IP developers about their expectations in order to avoid inappropriate use, likewise Tier-1 suppliers of controllers to OEMs may also develop using the SEooC model. The availability of a safety documentation package at IP level is therefore enabling throughout the value chain and so very important part of any IP offering.

## Functional safety becoming more of a requirement

Functional safety is moving away from a specialist requirement to become normality as the number of complex applications that rely on electronics is increasing, ranging from automotive to medical and industrial devices. Functional safety is growing in importance for SoC vendors as there is a realisation that a wide range of markets can benefit from highly reliable systems. By developing IP to rigorous standards and providing safety documents, ARM is simplifying the process for its silicon partners to meet safety standards. With its roots firmly in the realm of quality and robustness, work done to enable functional safety also has widespread benefits and is a catalyst for improved

quality and product resilience across the industry. It is a foundation upon which chip designers can develop systems to address the need for higher performance in the car; for driver safety, fuel efficiency, comfort, and in-car infotainment.

## Trademarks