Safety Assessment and Design of Dependable Cybercars: For today and the future

Arslan Munir, *Member, IEEE*

Abstract—Future generation of automobiles (also known as cybercars) will further increase the proliferation of electronic control units (ECUs) to contrive novel infotainment and distributed control applications. ISO 26262 is an automotive standard, which designates automotive safety integrity levels (ASILs) to indicate the criticality associated with a function. This article aims to provide a concise discussion on safety assessment and design of dependable cybercars considering the current state-of-the-art and future perspectives. We elaborate risk assessment and design of dependable cybercars as stipulated in ISO 26262. We classify failure modes in cybercars and the methods to assess these failures. We further elaborate on various hardware architectural metrics to assess safety and dependability of a cybercar design. Finally, we discuss various research challenges and future research directions for realizing safe and dependable cybercars.

Index Terms—Automotive, safety, dependability, failure modes, fault tolerance, electronic control unit, ISO 26262

1 Introduction and Motivation

Modern automobiles consist of more than 100 electronic control units (ECUs) to contrive various distributed control applications. Next generation of automobiles (also known as cybercars) will further increase the proliferation of ECUs as 90% of modernization in automobiles is due to electronic systems (perceived by Daimler Chrysler [1]). These innovative applications include x-by-wire systems, such as steer-by-wire, throttle-by-wire, brake-by-wire, and various driving assistance systems. Although computerized control of automotive systems offers various performance, comfort, and safety benefits; this computerized control also introduces vulnerabilities associated with electronic designs, such as process variation, electrostatic discharge, electromagnetic compatibility, and operational lifetime. Considering that the total number of motor vehicle fatalities in the United States from 1899 to 2013 is 3,613,732 [2], there has been an increasing emphasis on dependability and safety integration in automobiles.

To assist the design and production of safe automotive systems, International Organization for Standardization (ISO) has developed a functional safety standard, viz., ISO 26262 [3]. Although currently there are no legal certification requirements for adherence of automotive E/E systems with ISO 26262, automotive original equipment manufactures (OEMs) are obligated indirectly to produce safe products that are in compliance with ISO 26262 under *product liability* legislations. The product liability law holds accountable the manufacturers, suppliers, distributors, and retailers for the damages caused by their products [4]. Legally, a product is considered to be *safe* if the product conforms to the cutting-edge science and technology at the time of the product release. ISO 26262 is the contemporary state-of-the-art in automotive science and technology, and will continue to be regarded as such until this standard is replaced by a more leading-edge standard.

Incorporating safety and reliability in automotive designs is not only a legal obligation but also a moral commitment for automotive OEMs and suppliers as quoted by Werner von Siemens [5] (founder of Siemens Corporation): *"The prevention of accidents must not only be considered as a regulation by law, but as a matter of human commitment and economic reason"*. As electronic systems permeate into safety-critical functions; reliability, safety and hazard analysis of failure modes become imperative. A failure caused by a design error in safety-critical systems can lead to serious accidents ensuing damaging litigation costs, recall costs, and reputation loss. Significance of dependability assimilation in automotive electronics (semiconductor components) can be manifested with a simple example. Volkswagen Group produces more than 8 million vehicles per year with more than 40 billion integrated semiconductors. A semiconductor device failure rate of 0.1 parts per million (ppm) would result in 4000 failed vehicles per year due to semiconductor issues [6]. Although dependability integration in automobiles is indispensable, the dependability integration presents various challenges.

Fig. 1 portrays an overview of risk assessment and design of dependable cybercars. Harsh operating environments coupled with external noise and radiation render automotive electronic systems vulnerable to permanent, transient, and intermittent faults. While permanent faults can undermine or halt a system's correct functionality, soft errors induced by transient faults can remarkably decrease a system's availability. Addressing soft errors due to neutron strikes portends a substantial problem because sufficient shielding is exorbitantly costly. Intermittent faults oscillates between quiescent (i.e., component functions normally) and active (component malfunctions) states. Integration of safety and dependability in cybercars requires thorough understanding of different types of failures, failure

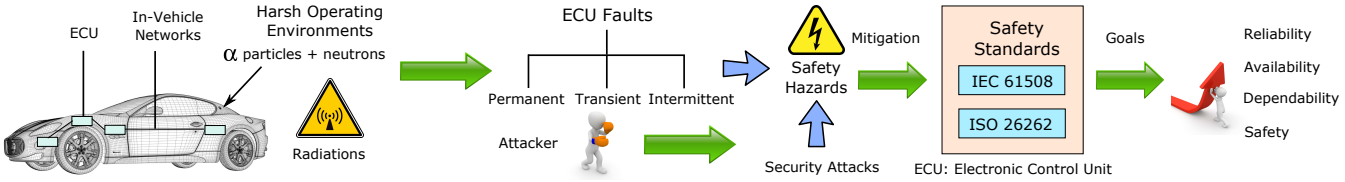


Fig. 1: Risk assessment and design of dependable cybercars.

modes, and safety standards. This article aims to provide a concise discussion on safety assessment and design of dependable cybercars considering the contemporary state-of-the-art and future perspectives. The main contributions of this article are as follows:

- Elaboration of automotive safety standards for cybercar design focusing on ISO 26262.
- Elucidation of different types of failures in automotive electronics including hardware failures, software failures, and soft failures.
- Characterization of failure modes and failure assessment in cybercars.
- Highlighting various research challenges and future research directions for designing safe and dependable cybercars.

The remainder of this paper is organized as follows. Section 2 discusses contemporary automotive safety standards focusing on ISO 26262. Section 3 elaborates failure and failure rate in context of automotive electronics. Failure modes in automotive systems are classified and assessed in Section 4. Section 5 discusses research challenges and future research directions for safe and dependable cybercars. Finally, conclusions are summarized in Section 6.

2 Safety and Safety Standards

Safety is one of the cardinal issues in the design of cybercars. We define safety with reference to two existing safety standards: (1) IEC 61508 [7], which is a functional safety standard for general electronics market developed by International Electrotechnical Commission (IEC); and (2) ISO 26262 [3], which is a functional safety standard for automobiles developed by ISO. IEC 61508 defines *safety* as exoneration from an unacceptable risk of harm or physical injury to the health of people either directly or indirectly as a result of impairment to the environment or property. ISO 26262 defines *functional safety*, which is a constituent of overall safety, as the freedom from an intolerable risk due to hazards caused by malfunction of E/E (i.e., electrical and/or electronic) systems.

Functional safety of a system is related to the system failure rate. All systems have some innate failure rate and it is practically impossible to design a system with zero failure rate. For each automotive application, there is some acceptable failure rate that does not lead to intolerable risk. The acceptable failure rates differ for different application and depend on the possibility for direct or indirect physical harm in case of system breakdown. The safety standards quantify the level of risk involved in case of failure for different applications into categories known as *safety integrity levels (SILs)*.

Since ISO 26262 targets automotive industry, our safety standards discussion focuses on ISO 26262. ISO 26262 [3] is an automotive industry standard designated for safety-related systems for series production passenger vehicles with a maximum gross vehicle mass up to 3500 kg and that are equipped with one or more E/E subsystems. ISO 26262 is not intended for E/E subsystems in special purpose vehicles (e.g., vehicles designed for drivers with disabilities). ISO 26262 identifies potential hazards caused by failure of E/E safety-related systems. The standard does not target hazards related to fire, smoke, shock, corrosion, radiation, heat, and similar hazards unless directly induced by impairment of E/E safety-related systems. ISO 26262 is derived from IEC 61508 [7] but is adapted to automotive industry. One of the major differences between IEC 61508 and ISO 26262 is that ISO 26262 deliberates controllability (defined below) whereas IEC 61508 does not.

ISO 26262 specifies guidelines to accomplish a functional safety evaluation and renders analysis procedures to determine automotive safety integrity levels (ASILs). An ASIL designates a function's/item's essential safety requisites for attaining an acceptable residual risk. ISO 26262 construes an *item* as a system or array of systems to realize a function at the vehicle level. Hazard analysis and risk assessment is an important constituent of ISO 26262 with the objective to classify hazards of an item/function and develop the safety goals (in terms of ASILs) in order to prevent or mitigate unacceptable risks and hazards. An ASIL is specified as one of the four levels: ASIL-A, -B, -C, or -D, where ASIL-D designates the most constrained and ASIL-A the least constrained level. The class quality management (QM) denotes no requirements with respect to ISO 26262.

ISO 26262 stipulates ASIL determination for each precarious event using estimation parameters: *severity* (S), probability of *exposure* (E), and *controllability* (C). The severity is determined by the potential damage and the exposure is estimated from the situation. The controllability is assigned depending on the ease or difficulty for the driver or the other road traffic participant to prevent an accident in the given scenario. ISO 26262 *functional safety concept* stipulates fundamental safety methods in terms of functional safety requirements. The functional safety concept addresses the following safety-specific mechanisms [8]:

- Fault detection and failure prevention.
- Transitioning to a safe state. A *safe state* is considered

to be a state that averts or alleviates a hazardous situation.
- Fault detection and driver warning (e.g., repair or stop request) to curtail the risk exposure time to a tolerable interval.
- Fault tolerance (FT) mechanisms that maintain the system in a safe state (with or without degradation), such that a fault does not cause directly the infringement of a safety objective.

ISO 26262 safety requirements structure is depicted in Fig 2. The safety requirements structure describes the safety-critical system's design process, which can be summarized in following three steps:
- System level safety concept, which includes definition of tolerable risk/ASILs, safety functions classification, and allocation of safety goals per function.
- Assignment of system level safety requirements to functional subsystems.
- Assignment of hardware and software requirements per component (e.g., ECUs) to implement a functional subsystem.

Fig. 2 shows that the hazard and risk analysis assessment is performed in the concept phase of the design. The safety goals and functional safety requirements are then assigned (in terms of ASILs) based on this hazard and risk analysis evaluation. Technical safety requirements are specified in the product development phase, which are then further designated to hardware and software components of the safety-related function. ISO 26262 requires establishing self control and safe state for the system during product development. The *self control* requires specification of means to detect failure, and then transitioning the system into a safe state. We point out that ASIL is calculated for a function and not for a physical system component. The hardware and software components that realize a given function inherit the ASIL associated with that function. A hardware and/or a software component can realize several functions with different ASILs, in which case the hardware and/or software component inherit the integrity level associated with the function with the highest ASIL. A technique called *ASIL decomposition* can lower the ASIL associated with hardware and software components under certain circumstances. ASIL decomposition, if possible, decomposes a component used to address a given safety goal into two independent components each with possibly lower ASIL but targeting the same original safety goal. ASIL decomposition can help lowering production costs as it is usually less costly to develop components with lower ASIL.

ISO 26262 also specifies minimum testing requirements for qualification of a hardware/software component depending on the component's ASIL. *Hardware qualification* of a component determines the component's suitability for the overall system and also assesses the component's failure modes. Hardware qualification requires testing the component

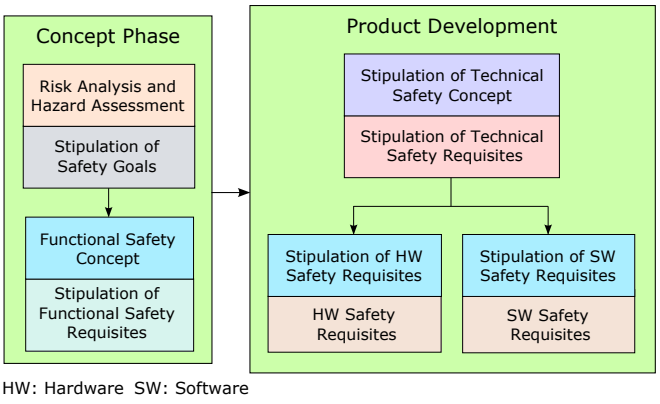


Fig. 2: ISO 26262 structure of safety requirements.

in different operational and environmental settings. Various numerical methods are leveraged to analyze the component's test results. The test results are then released in a qualification report in addition to the testing procedure, input criteria, and assumptions. *Software qualification* demands testing under normal operating settings with faults injected in the system to analyze the system's reaction to anomalous inputs. Safety-critical automotive systems must reciprocate appropriately to test scenarios and must subsist within designated safety bounds when subjected to different environmental and human inputs. ISO 26262 specifies testing requirements because high quality testing systems can ameliorate a product's quality, reliability, and performance. Compliance to the standard gives automobile OEMs and suppliers lucid guidelines for a safety-oriented design and testing, and enables them to assess risks and hazards of a system during early design phase, concept phase, or production phase. Industry estimates that a failure cost reduces by $10\times$ when an error is detected in the production phase rather than in the field and further reduces by $10\times$ if the error is detected in the design phase in lieu of production phase [9].

3 Failure and Failure Rate

The hardware and software component of automotive electronics constitute 50% each, however, the software component is growing due to economic considerations and short time-to-market. For example, the software code size for automotive electronics was 1.1 KB in 1980, 2 MB in 2000, and 10 MB in 2004. Studies reveal that hardware-related failures account for 30% to more than 60% of total failures in computer systems [10]. Software-related failures are the second largest cause of failure in computer systems with failure percentages ranging from 5% to 24% [10].

Reliability of a system depends on its failure rate. A component's/system's reliability is expressed by three common terms: failure rate, mean time to failure (MTTF), and failures in time (FIT). The *failure* rate represents the rate at which an individual component experiences faults. The failure rate at time t signifies the conditional probability that the component will fail in the time interval $[t + dt]$ given that the component has not failed

TABLE 1: Tolerable failure rate per hour λ_{SIL} corresponding to various SILs as per IEC 61508.

SIL	Low Demand Mode	High Demand Mode
1	$10^{-2} \leq \lambda < 10^{-1}$	$10^{-6} \leq \lambda < 10^{-5}$
2	$10^{-3} \leq \lambda < 10^{-2}$	$10^{-7} \leq \lambda < 10^{-6}$
3	$10^{-4} \leq \lambda < 10^{-3}$	$10^{-8} \leq \lambda < 10^{-7}$
4	$10^{-5} \leq \lambda < 10^{-4}$	$10^{-9} \leq \lambda < 10^{-8}$

until time t. The failure rate is constant for a system whose time to failure can be expressed by an exponential distribution. The MTTF of a component is the expected (mean) time to failure. One FIT means one failure per billion hours of operation, that is, 1 FIT = 1×10^{-9} failures per hour. Since MTTF and FIT of a component/system can be derived from its failure rate, this section discusses failure rate in detail.

Hardware Failures: Hardware failures occur due to faults in hardware components. Hardware failures in a component/system are regulated by fundamental processes (e.g., electrical, mechanical, thermal, radiation, chemical) and are initiated when the imposed stress due to these processes exceeds the mechanical strength of the component/system. Tolerable failure rates for a particular SIL depend on the usage (exposure) probability of the implemented safety function (i.e., a safety function which is more frequently used/exposed has a stronger requirement on the tolerable failure rate). Table 1 depicts the tolerable failure rate λ_{SIL} corresponding to various SILs both for low and high demand (continuous demand) functions as outlined in IEC 61508. Most automotive functions/applications are considered high or continuous demand. Safety awareness and its necessity in automotive systems has obligated component suppliers to receive SIL certifications. Many automotive suppliers are developing ECUs to meet automotive safety requirements and to comply with the functional safety norms, such as IEC 61508 and ISO 26262. For example, the Texas Instrument's TMS570F, which a lock step, dual-core ECU based on ARM Cortex-R4F, has been certified by Exida Consulting LLC to be IEC 61508 SIL-3 capable. ISO 26262 specifies tolerable random hardware failure rate λ_{ASIL} for ASIL-A, ASIL-B, ASIL-C, and ASIL-D as less than 10^{-6}/h, 10^{-7}/h, 10^{-7}/h, and 10^{-8}/h, respectively. In other words, ASIL-A, ASIL-B, ASIL-C, and ASIL-D can tolerate 1000 FIT, 100 FIT, 100 FIT, and 10 FIT, respectively, which are equivalent to SIL 2, SIL3, SIL3, and SIL4, respectively.

Software Failures: Software failures occur due to mistakes in capturing software requirements or during software development. In contrast to hardware, software has no random failure modes (Section 4) or wear-out phase. Moreover, the software failure probability does not increase with elapsed time instead the failure probability depends on the probability of occurrence of inputs/conditions necessary to trigger the failures.

Software integrity of safety-related functions can be ensured by methods similar to control systematic hardware failures (i.e., methods focusing on design, construction, and validation of the product). Software errors in safety-critical applications can be minimized by the programming language's and the compiler's adherence to international standards. In particular, the compiler and programming language for safety-critical applications need to have an unambiguous definition, be strongly typed, support detection of programming errors, and support limited language subsets as recommended by automotive software reliability associations.

Motor Industry Software Reliability Association (MISRA) [11] is a consortium formed by vehicle OEMs, component suppliers, and engineering consultancies to provide assistance to automotive industry in creation of safe and reliable software. MISRA has introduced MISRA-C, which is a subset of C programming language, for safety-critical applications. MISRA-C has been widely adopted by automotive, aerospace, medical, and industrial markets. Similarly, MISRA has introduced MISRA-C++ (in 2008), which is a set of rules for use of C++ in safety-critical systems and specifies a safe subset of C++ language.

Soft Errors and Soft Failures: A soft error, also known as single event upset (SEU), is construed as a transient portion of erroneous machine state. Since this type of fault does not reflect a device's permanent error, it is termed as soft or transient. A soft error in logic manifests when the outcome of a transient fault in logic reaches a storage element and is latched. A soft error in a memory element happens when adequate charge is produced to flip the value stored in the memory element. Soft errors are induced by high energy particle impacts, such as neutrons from cosmic rays and alpha particles from the radioactive decay of integrated circuit packages.

4 Failure Modes Classification and Assessment

Considering the proliferation of computerized control in modern automotive systems, it is imperative to classify and assess various failure modes in automotive electronics. This section discusses failure modes classification and assessment in automotive systems as stipulated by ISO 26262.

4.1 Failure Classification

ISO 26262 classifies failures in a functional safety system in three categories: (i) systematic failures, (ii) random failures, and (iii) dependent failures.

Systematic Failures: Systematic failures represent the failures in an item/function that are induced in a deterministic way during development, manufacturing, or maintenance. Systematic failures stem from a design or manufacturing flaw, which is a consequence of ignoring best practices. The systematic failure rate can be reduced through improvements in design, analysis, and verification processes.

Hardware Random Failures: Hardware random failures transpire unpredictably during the lifetime of a

hardware element and emanate from random defects innate to the process or usage conditions. Hardware random failures are engendered by permanent faults (e.g., stuck-at faults), transient faults (e.g., SEUs or soft errors), and/or intermittent faults (e.g., time dependent alternations). Designers emphasize detection and handling of random failures to avoid hazards due to these failures. In particular, techniques used to avoid random failures include redundancy, diagnosis, monitoring, and self-tests.

Dependent Failures: Dependent failure, also known as common-cause failure, is defined as the failure of two or more elements of an item emanating from a single root source or a distinct event. Dependent failures can be systematic or hardware random failures. The main causes of dependent failures include environmental conditions (e.g., humidity, pressure, temperature, vibration, corrosion), electromagnetic compatibility, stress due to a particular situation (e.g., aging), and failures of mutual external sources (e.g., input data and power supply). Dependent failures can be reduced by supervision of clock, power, temperature, and independent failure signaling.

4.2 Failure Modes Classification

Failure modes need to be defined for a hardware element in order to calculate hardware architectural metrics (Section 4.3). Fig. 3 depicts classification of failure modes of a hardware element. These fault modes are defined below according to ISO26262:

Safe Fault: A safe fault is a fault whose appearance does not increase considerably the probability of infraction of a safety goal.

Multiple Point Fault (MPF): An MPF is an individual fault that in conjunction with other independent faults causes a multiple point failure. An MPF can be either perceived, detected, or latent as defined below:

- A *perceived MPF* is a fault that is deduced by the driver within a specified time and is not identified by a safety mechanism.
- A *detected MPF* is a fault that is detected by a safety mechanism within a specified time to avert the fault from being latent.
- A *latent MPF* is a fault whose presence is neither identified by a safety mechanism nor recognized by the driver.

Single Point Fault (SPF): A SPF is a fault (in an element) that is not shielded by a safety structure and causes directly the infraction of a safety objective.

Residual Fault: A residual fault is a portion of a fault (in an element) that is not protected by existing safety mechanisms and the portion of the fault by itself causes the breach of a safety objective.

4.3 Failure Assessment via Hardware Architectural Metrics

ISO 26262 defines hardware architectural metrics to objectively assess hardware random failures.

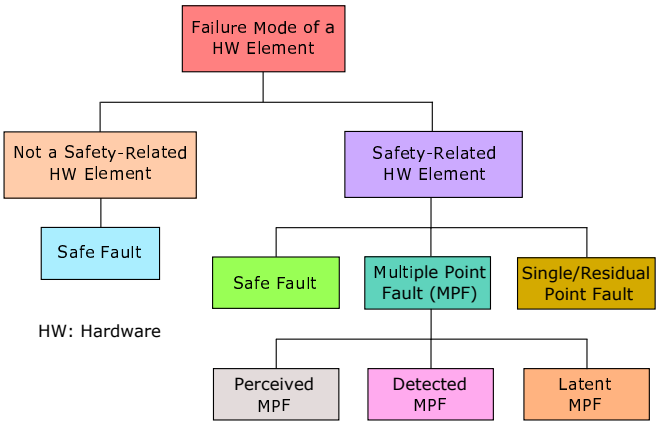


Fig. 3: Classification of failure modes of a hardware element (adapted from [12]).

The hardware architectural metrics are attestable, unequivocal, and accurate enough to distinguish between various architectures. These metrics support design assessment and determination of whether or not the incorporated safety mechanisms provide adequate protection to control hardware faults. ISO 26262 defines following hardware architectural metrics:

Single Point Fault (SPF) Metric: An SPF metric signifies the robustness of an item/function to the single point faults either by design or by coverage from safety procedures. A high SPF metric indicates that the hardware architecture (that implements the item/fuction) has a low proportion of single point faults and residual faults. The SPF metric is defined as [8]:

$$\text{SPF metric} = 1 - \frac{\sum_{\text{HW}_{\text{SR}}}(\lambda_{SPF} + \lambda_{RF})}{\sum_{\text{HW}_{\text{SR}}} \lambda} \quad (1)$$

$$= \frac{\sum_{\text{HW}_{\text{SR}}}(\lambda_{MPF} + \lambda_S)}{\sum_{\text{HW}_{\text{SR}}} \lambda} \quad (2)$$

where $\sum_{\text{HW}_{\text{SR}}}$ denotes safety-related hardware elements; λ_{SPF}, λ_{RF}, λ_{MPF}, and λ_S denote the failure rates associated with single point faults, residual faults, multiple point faults, and safe faults, respectively; and λ denotes the failure rate corresponding to all the hardware faults and is given by:

$$\lambda = \lambda_{SPF} + \lambda_{RF} + \lambda_{MPF} + \lambda_S \quad (3)$$

Latent Fault (LF) Metric: An LF metric reflects the robustness of an item/function against latent faults either by design (primarily safe faults), fault coverage via safety procedures, or by the driver's recognition of a fault's existence before the infraction of a safety objective. The LF metric is defined as:

$$\text{LF metric} = 1 - \frac{\sum_{\text{HW}_{\text{SR}}}(\lambda_{MPF_L})}{\sum_{\text{HW}_{\text{SR}}}(\lambda - \lambda_{SPF} - \lambda_{RF})} \quad (4)$$

$$= \frac{\sum_{\text{HW}_{\text{SR}}}(\lambda_{MPF_D} + \lambda_S)}{\sum_{\text{HW}_{\text{SR}}}(\lambda - \lambda_{SPF} - \lambda_{RF})} \quad (5)$$

where λ_{MPF_L} and λ_{MPF_D} denote the failure rates associated with latent MPFs and detected MPFs, respectively. λ is given by Eq. 3. Table 2 depicts target

TABLE 2: Single point fault metric and latent fault metric target values for various ASILs [3][13].

Faults metric	ASIL-A	ASIL-B	ASIL-C	ASIL-D
SPF metric	N/A	$> 90\%$	$> 97\%$	$> 99\%$
LF metric	N/A	$> 60\%$	$> 80\%$	$> 90\%$

values for SPF metric and LF metric for various ASILs as specified by ISO 26262.

Probabilistic Metric for Random Hardware Failures (PMHF): A PMHF is an absolute metric that quantifies the maximum probability of infraction of a safety objective due to random hardware failures and is represented by the average failure probability per hour over the lifetime of the item/function.

4.4 Hardware Architectural Metrics Calculation

Hardware architectural metrics can be used to determine if a designed system meets given ASIL requirements. For example, PMHF can be used in a four step procedure to determine a design's conformance to given ASIL requirements.

1. Overall failure rate estimation of hardware components comprising an automotive system.
2. Substituting the determined failure rates in the automotive system's model (e.g., Markov model).
3. Determining the system reliability from the model.
4. Comparison of calculated system's reliability with target values for a given ASIL (reliability values corresponding to a given ASIL can be determined assuming a failure rate distribution (e.g., exponential distribution)).

Alternatively, calculation of other hardware architectural metrics (i.e., SPF metric and LF metric) provides objective evidence that a given hardware design has achieved a safety goal as summarized in the following four steps [12]:

1. Failure rate estimation of SPF and latent MPF.
2. Diagnostic coverage estimation of the safety mechanism.
3. Calculation of SPF and LF metrics.
4. Comparison of calculated hardware architectural metrics with target values for a given ASIL.

These steps are described in the following subsections:

4.4.1 Failure rate estimation of SPF and latent MPF

ISO 26262 suggests that the failure rate λ of a hardware element can be determined in any of the following ways:

- Using a recognized industry source (e.g., IEC 61709, IEC TR 62380, MIL-HDBK-217F, NPRD95, EN 50129 Annex C, EN 62061 Annex D, MIL-HDBK-338, RAC FMD97, etc.).
- Utilizing statistics based on tests or field returns.
- Employing expert judgement founded on an engineering method.

The hardware architectural metrics calculation also requires determining failure rates for different failure modes of a hardware element. Practically, hardware architectural metrics can be determined by estimating failure rates for only a few failure modes of the hardware element, which are λ_{SPF}, λ_{RF}, and λ_{MPF_L}. We use following equations to determine these failure rates:

$$\lambda_{SPF} + \lambda_{RF} = \lambda \cdot (1 - c) \tag{6}$$

where c denotes the diagnostic coverage of a safety mechanism for SPFs.

$$\lambda_{MPF_L} = \lambda \cdot (1 - c_l) \tag{7}$$

where c_l denotes the diagnostic coverage of a safety mechanism for latent MPFs.

4.4.2 Diagnostic coverage estimation of safety mechanism

Diagnostic coverage c of a safety mechanism depends on hardware/software techniques employed by the mechanism. ISO 26262 specifies minimum diagnostic coverage requirements for a safety function. For example, ASIL-B, ASIL-C, and ASIL-D safety functions require low ($> 60\%$), medium ($> 90\%$), and high ($> 99\%$) diagnostic coverage [5]. Diagnostic coverage for different fault detection techniques varies. For example, self-tests by software provide low diagnostic coverage, watchdogs provide medium diagnostic coverage, whereas comparators and majority voters provide high diagnostic coverage. Similarly, one bit redundancy in random access memory (RAM) can attain a low diagnostic coverage whereas error detection codes and block replication can provide comparatively higher diagnostic coverage for RAM errors.

4.4.3 Calculation of SPF and LF metrics

After determining failure rates of a hardware element and diagnostic coverage of the safety mechanism, we can obtain SPF and LF metrics using Eq. 1 and Eq. 4, respectively.

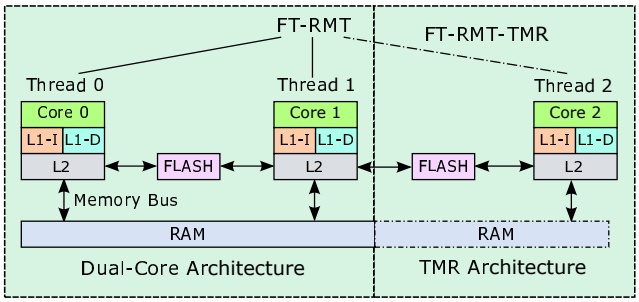
4.4.4 Comparison of calculated hardware architectural metrics with target ASIL values

To objectively claim the compliance of a product (i.e., incorporating a safety mechanism using a given hardware element) with ISO 26262, the determined hardware architectural values are compared with target ASIL values of the implemented safety function.

5 Research Challenges and Future Research Directions

Although efforts have been made towards development of safety mechanisms and safety standards for automobiles, there remains various challenges and issues related to safety and dependability of cybercars. In this section, we discuss several research challenges and future research directions for realizing safe and dependable cybercars.

Soft Error Rates: There has been an increase in SERs due to modern transistor and device scaling trends. Device scaling permits lower operating voltages that can offer energy savings but on the other hand decreases the energy needed to generate a voltage pulse at a



TMR: Triple Modular Redundancy RMT: Redundant Multi-Threading
FT-RMT: Fault-Tolerance by RMT RAM: Random Access Memory
FT-RMT-TMR: Fault-Tolerance by RMT on a TMR Architecture
L1-I: Level One-Instruction Cache L2: Level Two Cache
L1-D: Level One-Data Cache

Fig. 4: Dual-core and triple-core FT architectures.

logic gate's output or to flip the value stored within a sequential element. Hence, low energy particle strikes that did not present an issue in previous technology generations could engender soft errors in current and subsequent technology generations. Furthermore, the rate of particle strikes escalates exponentially with the decrease in the particles' energy levels. Consequently, the rate of particle strikes that can possibly impact the microprocessors' logical operations increases significantly with each new process technology. To minimize the impact of SERs on availability of safety-critical automotive systems, various FT techniques can be leveraged, such as acceptance tests, FT by redundant multi-threading (FT-RMT), N-version programming, and checkpointing.

ECU Architectures: To meet the dependability and performance requirements stipulated by automotive standards, automotive OEM suppliers have developed various paradigms for novel ECU architectures: dual-core architecture, lock-step dual processor architecture, loosely-synchronized dual processor architecture, and triple modular redundant (TMR) architecture. Fig. 4 depicts typical dual-core and TMR architectures that consist of two and three processor cores, respectively, a level one-instruction cache (L1-I), a level one-data cache (L1-D), and a level two cache (L2) per core on a single chip. The processor cores are connected via memory buses to a FLASH memory and a random access memory (RAM), also known as main memory. The dual-core architecture permits FT (error detection only) by FT-RMT and the TMR architecture permits FT (error detection plus single error correction) by FT-RMT-TMR. In the FT-RMT, two independent threads run on the two processor cores and an error is detected on a mismatch between the two threads' outputs. In the FT-RMT-TMR, a majority voter compares the outputs of the three threads and chooses the majority voted output. The design of novel dependable ECU architectures is a promising research area.

Advanced Driver Assistance Systems: Safety of automobiles can be enhanced by leveraging advanced driver assistance systems (ADAS). Some of the ADAS features include automated lighting, adaptive cruise control, automated braking, traffic warnings, and lane departure warning system. Safety of automobiles can be further reinforced by leveraging radar, lidar, and vehicle-2-X (V2X) communication technologies in ADAS. The integration of these new technologies in ADAS can help in detecting road hazards, such as broken roads, bumps and potholes, and obstacles, even in hazardous environmental conditions (e.g., fog, snow and sand storms). The realization of these additional features in ADAS will require advancements in computer vision and pattern recognition techniques.

Real-time Constraints: Many of the automotive embedded systems have stringent real-time constraints. The challenge is to meet the strict safety and dependability requirements of these automotive systems without violating hard real-time constraints. Specifically, the added redundancy in an FT design should not impair the automotive system's real-time performance. A prerequisite for providing FT is fault/error-detection. *Error detection latency* denotes the time passed between the happening of an error (due to a bug or external interference) and its detection at an observable point in the program. Long error detection latencies can result in missed real-time deadlines for safety-critical automotive systems. Hence, timely detection or masking of soft errors is vital for real-time constrained applications. There is a need to develop dependability methodologies that would enable quick error detection and correction. For instance, an optimized combination of *comparison-points* (i.e., insertion of comparison instructions in a program to detect faults) and *lightweight checkpointing* can help to meet the application's real-time constraints even in the presence of faults.

Security: Modern automobiles provide several interfaces, such as on-board diagnostics port (OBD-II), entertainment systems (e.g., CD, USB, iPod), and short range or long range wireless access, that provide direct or indirect access to an automobile's internal networks. Since messages are transmitted in plain-text format over in-vehicle networks (e.g., CAN, CAN FD, and FlexRay), intruders may be able to gain access and even alter these messages creating security threats. Cyber-physical aspects of modern automobiles directly couple security vulnerabilities to cybercars physical safety and dependability. The research challenge is to integrate security primitives (i.e., confidentiality, integrity, and authentication) over in-vehicle networks without violating real-time constraints imposed by the response time of automotive applications.

Energy Consumption: The challenge in the design of safe and dependable cybercars is integration of security and dependability primitives while minimizing energy consumption. The dependability methodologies that integrate *dynamic voltage and frequency scaling (DVFS)* and *lightweight checkpointing* can help in reducing energy consumption. The DVFS controller can help determining the operating voltage and frequency of ECUs to minimize energy consumption while ensuring

that real-time constraints are satisfied even in the presence of faults. The checkpointing can reduce the energy overhead because of reduced re-computation time in the presence of faults as the correct operating state can be restored from a saved checkpoint. Further research endeavors in energy-efficient integration of safety and dependability primitives are required to reduce emissions, pollution, and carbon footprint, and to provide greater fuel-efficiency for combustion engine vehicles and longer battery lifer for hybrid and electric vehicles.

Scalability: Scalability analysis assesses system's performance for a projected addition of new components/messages later in the design process. Scalability considerations are paramount in the design of safe and dependable automotive systems. The dependability and security approaches for automotive embedded systems must be scalable as the number of ECUs and messages on the in-vehicle networks increases.

Cost: Cost of electronic components in automobiles is increasing as majority of the innovation in automobiles is due to electronic systems. For example, automotive electronics cost 5% of the total car cost in 1970, 10% in 1980, 35% in 2010, and 50% in 2030 (projected) [14]. Enhancing safety, reliability, and availability at a minimal additional cost is a leading challenge in automotive embedded systems design. Since automotive systems have stringent cost constraints, the additional cost for safety and dependability integration must be curtailed by incorporating only as much dependability as required and not more.

Modeling and Tools: Most of the existing reliability and dependability analysis tools are general-purpose and do not provide integrated models for dependability analysis of automotive embedded systems. There is a need for developing novel modeling approaches and tools that can provide safety, hazard, availability, and performability analysis for automotive systems while considering the effects of electronics quality grade, temperature, and design lifetime.

6 Conclusions

In this article, we elaborate design of dependable and safe cybercars considering the current state-of-the-art and future perspectives. Based on the contemporary automotive standard (ISO 26262), the article summarizes structure of safety requirements including automotive safety integrity levels (ASILs) to signify a function's essential safety requirements and measures for avoiding an unreasonable residual risk. We discuss classification of automotive system failures in different categories, such as systematic failures, random failures, and dependent failures. We further elaborate on hardware architectural metrics (e.g., single point fault metric, latent fault metric, and probabilistic metric for random hardware failures) to assess safety and dependability of a cybercar design. Finally, we highlight some of the pressing research challenges and future research directions for designing safe and dependable cybercars.

Acknowledgments

This work was supported by the National Science Foundation (NSF) (NSF-CRII-CPS-1564801). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] F. Simonot-Lion, "The Design of Safe Automotive Electronic Systems: Some Problems, Solutions, and Open Issues," in *Keynote Presentation in IEEE Symposium on Industrial Embedded Systems (IES)*, Antibes Juan-Les-Pins, France, October 2006.

[2] Wikipedia, "List of Motor Vehicle Deaths in U.S. by Year," July 2013. [Online]. Available: http://en.wikipedia.org/wiki/List_of_motor_vehicle_deaths_in_U.S._by_year

[3] ISO26262, "Road vehicles – Functional safety," April 2013. [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=43464

[4] LexisNexis, "Products Liability," 2013. [Online]. Available: http://www.lexisnexis.com/lawschool/study/outlines/html/torts/torts17.htm

[5] C. Ebert and D. Kanth, "Functional Safety with ISO 26262: Principles and Practice," in *Vector Webinar*, 2012. [Online]. Available: http://www.vector.com/portal/medien/vector_consulting/publications/FunctionalSafety_Webinar_EN.pdf

[6] A. Aal and T. Polte, "On Component Reliability and System Reliability for Automotive Applications," in *Proc. of IEEE International Integrated Reliability Workshop Final Report (IIRW)*, South Lake Tahoe, California, October 2012, pp. 168–170.

[7] IEC61508, "Functional Safety and IEC 61508," April 2013. [Online]. Available: http://www.iec.ch/functionalsafety/

[8] M. Bellotti and R. Mariani, "How Future Automotive Functional Safety Requirements will Impact Microprocessor Design," *Elsevier Microelectronics Reliability*, vol. 50, no. 9-11, pp. 1320–1326, November 2010.

[9] National_Instruments, "What is the ISO 26262 Functional Safety Standard," February 2012. [Online]. Available: http://www.ni.com/white-paper/13647/en

[10] B. Schroeder and G. A. Gibson, "A Large-Scale Study of Failures in High-Performance Computing Systems," in *Proc. of IEEE International Conference on Dependable Systems and Networks (DSN)*, Philadelphia, Pennsylvania, June 2006.

[11] MISRA, "Motor Industry Software Reliability Association," April 2013. [Online]. Available: http://www.misra.org.uk/MISRAHome/tabid/55/Default.aspx

[12] S.-H. Jeon, J.-H. Cho, Y. Jung, S. Park, and T.-M. Han, "Automotive Hardware Development According to ISO 26262," in *Proc. of 13th International Conference on Advanced Communication Technology (ICACT)*, Phoenix Park Gangwon-Do, South Korea, February 2011, pp. 588–592.

[13] C. Madritsch, "ISO 26262 Product Development - Hardware and Software Level," 2012. [Online]. Available: http://ext02.fh-kaernten.at/rts/intern/downloads/AutomotiveSystems/ISO26262HardwareandSoftwareDesign.pdf

[14] S. P. Nelson, "Challenges and Technologies: The Human Friendly Vehicle in 2030 and Beyond," in *NXP*, 2009. [Online]. Available: http://www.nxp.com/files/training_pdf/VFTF09_AA106.pdf

Arslan Munir is an assistant professor in the Department of Computer Science and Engineering at the University of Nevada, Reno. His research interests include embedded and cyber-physical systems, computer architecture, multicore, parallel computing, fault tolerance, and computer security. Munir has a PhD in electrical and computer engineering from the University of Florida, Gainesville. He's a member of IEEE. Contact him at arslan@unr.edu.