Fault Handling Approaches on Dual-Core Microcontrollers in Safety-Critical Automotive Applications

Eva Beckschulze, Falk Salewski, Thomas Siegbert, and Stefan Kowalewski

Embedded Software Laboratory, RWTH Aachen University, Germany
surname@cs.rwth-aachen.de

Abstract. The number of safety-critical applications is increasing in the automotive domain. Accordingly, requirements given by recent safety standards have to be met in these applications. These requirements include a demonstration of sufficient measures for the handling of permanent and transient hardware faults. Moreover, a consideration of software faults is required. In this work, approaches based on dual-core microcontrollers are investigated with respect to their fault handling capabilities. Therefore, *function monitoring architectures* that are based on a supervision of the implemented function and *generic architectures*, which monitor the hardware executing the application, are compared. This comparison is then further illustrated by an application example. Summarizing, both approaches come with their specific advantages and disadvantages, which should be considered during the development of the functional safety concept.

1 Introduction

Modern automobiles include an increasing amount of functionalities and most of these functionalities are implemented in software to allow flexible and complex applications. Faults in most of these functions could lead directly or indirectly to serious accidents which makes the majority of functions implemented in today's automobiles safety-critical. Most popular examples are driver assistance systems for stability control and crash avoidance, which typically require access to at least the brake system. However, even comparably simple applications as the electronic locking of the steering wheel require extensive safety measures, as a malfunction easily results in an accident.

The consideration of safety aspects in automobiles is complicated by two major aspects. First of all, most of these systems are real-time systems which require a completed computation of tasks before a given deadline. This requirement includes the execution of all required safety measures. The second aspect is that high requirements for low costs and low power lead to controllers with restricted resources. These requirements result in restricted memory sizes and computation power, but also in the need to apply general purpose devices whenever possible.

T. Margaria and B. Steffen (Eds.): ISoLA 2008, CCIS 17, pp. 82–92, 2008.

The advent of dual-core microcontrollers might help to meet the mentioned challenges. Although these devices are not yet established in the automotive domain, different approaches to use them in safety critical applications were proposed already. This paper aims to compare these approaches for safety-critical applications with a safety integrity level of ASIL C (automotive safety integrity level according to ISO WD 26262 [6]). Therefore, the requirements for such an application are presented briefly in the following Section 2. Next, known approaches with dual-core microcontrollers are presented and compared in Section 3. Then, a dual-core approach is applied on an example automotive application presented in 4. Finally, a conclusion of these investigations is given in Section 5.

2 Requirements for ASIL C Application

Specific safety requirements have to be considered in safety-critical systems. In this regard, it is important that safety is a system property [10]. Thus, it has to be made sure that the **combination** of hardware and software never leads to an unsafe state. This property is typically achieved by implementing a sufficient *safety function*. A safety function is responsible for the detection and the handling of all faults which could lead to unsafe states of the overall system. One form of fault handling is to shut down the system as soon as a critical fault is detected (*fail-silent system*). Another option is to try a form of fault recovery. This recovery could include a simple reset of the system (could mitigate transient hardware faults and some software faults) or a more fine grained recovery (e.g. to defined recovery point in the system). During recovery, the actuators have to be put into a safe state to prevent potential hazards. Alternatively, the outputs might remain in their current state if fault handling can be achieved in a time shorter than the so-called *fault-tolerant time span* [6]. A disadvantage of these approaches is that the system cannot perform its service while the faults are handled. This disruption might be not acceptable for safety-critical systems that require permanent service (e.g. drive-by-wire system) and do not allow sufficiently fast recovery. A combination of two fail-silent units to one *fail-operational system* is one solution to this problem (see e.g. [15]).

For the automotive domain, a specific safety standard, namely the standard ISO WD 26262 [6] is currently developed[1]. The standard requires a comprehensive safety analysis, in which potential hazards are determined. These hazards are rated according to so-called automotive safety integrity levels (ASIL). They range from ASIL A to ASIL D with the latter representing the most demanding level. For each hazard, a safety goal is formulated that has to be assured by a suitable safety concept.

For applications rated as ASIL C, specific safety requirements are given in the ISO WD 26262. For the hardware parts of such a system, a sufficient handling of possible hardware faults has to be shown by the application of *fault metrics* and *coverage criteria* introduced in this standard. Accordingly, single point faults (faults that alone could violate a safety goal) are only permitted, if their risk

[1] As the standard is still a working draft, contents presented here might still change.