Chapter 2

Synthesis of Flexible Fault–Tolerant Schedules for Embedded Systems with Soft and Hard Timing Constraints

Viacheslav Izosimov
Semcon AB, Sweden

Paul Pop
Technical University of Denmark, Denmark

Petru Eles
Linköping University, Sweden

Zebo Peng
Linköping University, Sweden

ABSTRACT

In this chapter, the authors discuss an approach to the synthesis of fault-tolerant schedules for embedded applications with soft and hard real-time processes. The hard processes are critical and must always complete on time. A soft process can complete after its deadline and its completion time is associated with a value function that characterizes its contribution to the quality-of-service or utility of the application. Deadlines for the hard processes must be guaranteed even in the case of faults, while the overall utility should be maximized. Process re-execution is employed to recover from multiple transient and intermittent faults.

The authors present a quasi-static scheduling strategy, where a set of schedules is synthesized off-line and, at run time, the scheduler will select the appropriate schedule based on the occurrence of faults and the actual execution times of processes. The presented scheduling approach is further extended with preemption of soft and hard real-time processes, where the corresponding algorithms determine off-line when to preempt and when to resurrect processes.

DOI: 10.4018/978-1-60960-212-3.ch002
The authors also present evaluation of the schedule synthesis heuristics with and without preemption using extensive experiments and a real-life example.

INTRODUCTION

Fault-tolerant embedded real-time systems have to meet their deadlines and function correctly in the worst-case and with the presence of faults. Such systems are usually designed for the worst-case, which often leads to overly pessimistic solutions. Design of fault-tolerant embedded real-time systems for the average case, addressed in this chapter, is a promising alternative to the purely worst-case-driven design. It is important to emphasize that the generated designs have to be safe, i.e. all hard deadlines are met, even in the worst-case execution scenarios and when affected by faults.

Faults can be permanent (i.e. damaged micro-controllers or communication links), transient, or intermittent. Transient and intermittent faults (also known as “soft errors”) appear for a short time and can be caused by electromagnetic interference, radiation, temperature variations, software “bugs”, etc. Transient and intermittent faults, which we will deal with in this chapter, are the most common and their number is increasing due to greater complexity, higher frequency and smaller transistor sizes (Izosimov, 2009). We will refer to both transient and intermittent faults as “transient” faults since they manifest themselves similar from fault tolerance point of view.

Real-time systems have been classified as hard real-time and soft real-time systems. For hard real-time processes, failing to meet a deadline can potentially have catastrophic consequences, whereas a soft real-time process retains some diminishing value after its deadline. Traditionally, hard and soft real-time systems have been scheduled using very different techniques (Kopetz, 1997). However, many applications have both hard and soft timing constraints (Buttazzo, & Sensini, 1999), and therefore researchers have proposed techniques for addressing mixed hard/soft real-time systems (Buttazzo, & Sensini, 1999; Davis, Tindell, & Burns, 1993; Cortes, Eles, & Peng, 2004). Particularly, Cortes et al. (2004) have developed a design approach for multiprocessor embedded systems composed of soft and hard processes. A number of quasi-static scheduling heuristics has been proposed such that the overall utility of soft processes is maximized while deadlines of hard processes are satisfied. However, neither Cortes et al. (2004) nor any other of the above mentioned work on mixed soft and hard real-time systems has addressed fault tolerance aspects. In this chapter, thus, we present a novel approach to design fault-tolerant mixed soft/hard real-time systems. The approach is generic and can be applied on a variety of embedded systems, in particular, on systems-on-chip (SoC) used in factory automation, telecommunication and medical equipment and, last but not least, automotive electronics.

In the past and current research work, fault tolerance aspects have been traditionally addressed separately for hard and for soft real-time systems. In the context of hard real-time systems, for example, researchers have shown that schedulability can be guaranteed for online scheduling with fault tolerance (Han, Shin, & Wu, 2003; Liberato, Melhem, & Mosse, 2000; Ying Zhang, & Chakrabarty, 2006). However, such approaches lack the predictability required in many safety-critical applications, where static off-line scheduling is the preferred option for ensuring both the predictability of worst-case behavior, and high resource utilization. Thus, researchers have proposed approaches for integrating fault tolerance into the framework of static scheduling. A heuristic for combining together several static schedules in order to mask fault patterns through replication is proposed in (Pinello, Carloni, &