Chapter 3
Optimizing Fault Tolerance for Multi-Processor System-on-Chip

Dimitar Nikolov
Linköping University, Sweden

Mikael Väyrynen
Linköping University, Sweden

Urban Ingelsson
Linköping University, Sweden

Virendra Singh
Indian Institute of Science, India

Erik Larsson
Linköping University, Sweden

ABSTRACT

While the rapid development in semiconductor technologies makes it possible to manufacture integrated circuits (ICs) with multiple processors, so called Multi-Processor System-on-Chip (MPSoC), ICs manufactured in recent semiconductor technologies are becoming increasingly susceptible to transient faults, which enforces fault tolerance. Work on fault tolerance has mainly focused on safety-critical applications; however, the development of semiconductor technologies makes fault tolerance also needed for general-purpose systems. Different from safety-critical systems where meeting hard deadlines is the main requirement, it is for general-purpose systems more important to minimize the average execution time (AET). The contribution of this chapter is two-fold. First, the authors present a mathematical framework for the analysis of AET. Their analysis of AET is performed for voting, rollback recovery with checkpointing (RRC), and the combination of RRC and voting (CRV) where for a given job and soft (transient) error probability, the authors define mathematical formulas for each of the fault-tolerant techniques with the objective to minimize AET while taking bus communication overhead into account. And, for a given number of processors and jobs, the authors define integer linear programming models that minimize AET including communication overhead. Second, as error probability is not known at design time and

DOI: 10.4018/978-1-60960-212-3.ch003
it can change during operation, they present two techniques, periodic probability estimation (PPE) and aperiodic probability estimation (APE), to estimate the error probability and adjust the fault tolerant scheme while the IC is in operation.

1. INTRODUCTION

The rapid development in semiconductor technologies has enabled fabrication of integrated circuits (ICs) that can include multiple processors, referred to as multi-processor system-on-chips (MPSoCs). The drawback of the semiconductor development is that ICs are becoming increasingly sensitive to soft (temporary) errors that manifest themselves when the IC is in operation (Kopetz, Obermaisser, Peti, & Suri, 2004), (Sosnowski, 1994). The soft error rate has increased by orders of magnitude compared with earlier technologies, and the rate is expected to grow in future semiconductor technologies (Borel, 2009). It is becoming increasingly important to consider techniques that enable error detection and recover from soft errors (Borel, 2009), (Borkar, 1999), (Mukherjee, 2008). In this chapter we focus on fault-tolerant techniques addressing soft errors (Borel, 2009) (Chandra & Aitken, 2008).

Fault tolerance has been subject of research for a long time. John von Neumann introduced already in 1952 a redundancy technique called NAND multiplexing for constructing reliable computation from unreliable devices (von Neuman, 1956). Significant amount of work has been produced over the years. For example, researchers have shown that schedulability of an application can be guaranteed for pre-emptive on-line scheduling under the presence of a single transient fault (Bertossi & Mancini, 1994), (Burns, Davis, & Punnekat, 1996), (Han, Shin, & Wu, 2003), (Zhang & Chakrabarty, 2006). Punnekat et al. assume that a fault can adversely affect only one job at a time (Punnekat, Burns, & Davis, 2001). Kandasamy et al. consider a fault model which assumes that only one single transient fault may occur on any of the nodes during execution of an application (Kandasamy, Hayes, & Murray, 2003). This model has been generalized in the work of Pop et al. to a number k of transient faults (Pop, Izosimov, Eles, & Peng, 2005). Most work in the area of fault tolerance has focused on safety-critical systems and the optimization of such systems (Al-Omari, Somani, & Manimaran, 2001), (Bertossi, Fusiello, & Mancini, 1997), (Pop, Izosimov, Eles, & Peng, 2005). For example the architecture of the fighter JAS 39 Gripen contains seven hardware replicas (Alstrom & Torin, 2001). For a general-purpose system (non safety-critical system), for example a mobile phone, redundancy such as the one used in JAS 39 Gripen, seven hardware replicas, is too costly. For general-purpose systems, the average execution time (AET) is more important than meeting hard deadlines. For example, a mobile phone user can usually accept a slight and temporary performance degradation, so that error-free operation is ensured.

There are two major drawbacks with existing work. First, there is for general purpose systems no framework that can analyze and guide to what extent to make use of fault tolerance while taking cost (performance degradation and bus communication) into account. Second, approaches depend on a known error probability; however, error probability is not known at design time, it is different for different ICs, and it is not constant through the lifetime of an IC due to for example aging and the environment where the IC is to be used (Cannon, Kleinosowski, Kanj, Reinhardt, & Joshi, 2008), (Karnik, Hazucha, & Patel, 2004), (Koren & Krishna, 1979), (Lakshminarayanan, 1999).

In this chapter, we address fault tolerance for general-purpose systems. We assume given is an MPSoC, which has a few spare processors and a few jobs that are critical and require fault toler-