Microprocessors and Microsystems 39 (2015) 686–692

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

On the characterization and optimization of system-level vulnerability for instruction caches in embedded processors

Shuai Wang*, Guangshan Duan

State Key Laboratory of Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, Jiang Su, China

ARTICLE INFO

Article history:
Available online 8 October 2015

Keywords:
Embedded processors
Instruction cache
Reliability
Soft errors
Vulnerability factor

ABSTRACT

With continuous scaling down of the semiconductor technology, the soft errors induced by energetic particles have become an increasing challenge in designing current and next-generation reliable microprocessors. Due to their large share of the transistor budget and die area, cache memories suffer from an increasing vulnerability against soft errors. Previous work based on the vulnerability factor (VF) analysis proposed analytical models to evaluate the reliability of on-chip data and instruction caches. However, we have no possession of a system-level study on the vulnerability of instruction caches. In this paper, we propose a new analytical model to estimate the system-level vulnerability factor for on-chip instruction caches in embedded processors. In our model, the error masking/detection effects in instructions based on the Instruction Set Architecture (ISA) are studied. Our experimental results show that the self-error-masking/detection in instructions will reduce the VF of the instruction caches compared to the previous study. We also exemplify our design methodology by proposing several optimizing schemes to improve the reliability. Benchmarking is carried out to demonstrate the effectiveness of our vulnerability model and optimization approach, which can provide an insightful guidance for the future reliable instruction cache and ISA design.

1. Introduction

Ionizing radiation induced soft errors in semiconductor memories have been recognized for a long time as a major reliability issue in electronic systems [2,3]. Due to their large share of the transistor budget and die area, on-chip caches suffer from a significantly higher soft error rate (SER) than other on-chip components at the current and future technologies [4]. Incorrect data values/instructions once read out from the data/instruction caches may crash the subsequent computation/communication, external memory, or storage systems, leading to overall system failures or program inaccuracy. As a critical requirement for reliable computing [5], protecting the information integrity in cache memories has captured a wealth of research efforts [5–16].

Recent work has made progress towards the cache vulnerability analysis and reliability optimization based on the analysis [6,12,13,15,17–19]. For example, early write-back schemes [9,15,17] were proposed to reduce the vulnerability factor (VF) of dirty cachelines in a write-back data cache. Wang et al. [13] proposed a clean cacheline invalidation (CCI) scheme in data and instruction caches to improve their reliability. Information redundancy is another fundamental approach in building reliable memory structures. Various coding schemes, such as the parity and ECC codings, are used to enhance information integrity in latches, register files, and on-chip caches, providing different levels of reliability at different performance, energy, and hardware costs. Some recent work [20–22] focuses on the soft error reliability in multi or heterogeneous core environment. Another form of information redundancy is to maintain redundant copies of the data items in the cache memories [10,23]. In these schemes, the cachelines in the data array are duplicated when they are brought into L1 caches on read/write misses or on write operations. However, maintaining redundant copies of cachelines presents great challenges to the effective bandwidth and energy dissipation of caches [5,23]. Note that soft errors in memory structures are not related to the correctness of the design. Therefore, they cannot be captured by formal verification or testing. Furthermore, soft errors are extremely difficult to predict due to the random nature of their occurrences, which makes cost-effective reliable processor designs against soft errors an increasing challenge. In [24], Mukherjee et al. proposed an architectural vulnerability factor (AVF) for reliability quantification based on whether each bit during execution will affect the final system output. To provide an accurate upper bound estimation for AVF, a temporal vulnerability factor (TVF) was proposed in [13] for both data and instruction cache vulnerability characterization.

* Corresponding author. Tel.: +862589680908.
E-mail address: swang@nju.edu.cn (S. Wang).

http://dx.doi.org/10.1016/j.micpro.2015.09.011
0141-9331/© 2015 Elsevier B.V. All rights reserved.

Despite the fact that previous work has conducted some studies on the vulnerability factor characterization of the on-chip caches [12,13,15,17,25], we have no possession of a system-level vulnerability study on instruction caches against soft errors, especially for embedded processors. In this paper, we first analyze the vulnerability of the instruction cache in the embedded processor based on the previously proposed TVF [13] model. After the detailed study on the error masking/detection effects of the Instruction Set Architecture (ISA) to the instruction caches, we propose our System-level Instruction Cache Vulnerability Factor (SICVF) model. In order to further improve the reliability, we proposed several schemes to reduce the system vulnerability of the instruction cache, which can reduce the SICVF to 5.8% compared to the original 20.7%, with negligible performance overheads. The experimental results confirm that our study should be able to provide enough insight into the instruction cache reliability issues, which can be taken advantage of to design highly cost-effective reliable embedded processors.

The rest of the paper is organized as follows. The next section discusses the background and related work. Section 3 describes instruction cache vulnerability characterization based on the TVF model. In Section 4, we study different error masking/detection effects of the instruction set architecture to instruction caches and propose our new system-level instruction cache vulnerability factor model. Optimizing schemes to reduce the vulnerability of the instruction cache are proposed in Section 5. The experimental setup and evaluation are presented in Section 6. Section 7 concludes this work.

2. Background and related work

Soft Error Rate (SER) is an error rate metric for the system vulnerability due to soft errors. Failures in Time (FIT) is another widely-used error rate metric, which is inversely proportional to Mean Time to Failure (MTTF). The FIT rate of a component or a system is the number of failures it incurs over one billion (10^9) hours. One of the advantages of using the FIT metric is that the FIT rates can be added in an intuitive fashion. Therefore, the FIT rate of a system can be calculated according to the following equation.

$$FIT_{System} = \sum_i FIT_{Components} \tag{1}$$

Architectural Vulnerability Factor (AVF) [24] is a recently developed metric that provides insight into the structural vulnerability to soft errors. AVF can be used to scale a raw FIT rate into an effective FIT rate. The effective FIT can be calculated as follows:

$$FIT_{effective} = FIT_{raw} \times AVF \tag{2}$$

In [13], a Temporal Vulnerability Factor (TVF) was proposed to analyze the vulnerability of both on-chip data and instruction caches against soft errors. Different from AVF, TVF captured the upper bound of the cache vulnerability factor. However, TVF did not provide a system-level characterization on the cache vulnerability. Haghdoost et al. [26] extended the TVF model to estimate the system-level vulnerability factor of both write-through and write-back data caches by taking account of the read frequency and ALU masking. In this paper, we propose a system-level vulnerability model for instruction caches based on the error masking/detection effects of the instruction set architecture (ISA) for embedded processors. Compared to TVF, which only considered the temporal vulnerability within the instruction cache itself, our System-level Instruction Cache Vulnerability Factor (SICVF) also takes into account the error masking/detection effects after the instructions being fetched out of the instruction cache and during the execution. The idea is based on that the error bit in an instruction may be masked or detected due to the design of the ISA.

Due to the power and area constraints, the reliable embedded processor design differs from those in the high-performance processor domain [27,28]. Traditional fault-tolerant techniques like triple-modular redundant (TMR) may be too costly for embedded systems [29]. Therefore, the major challenge in reliable embedded processor design is maintaining the target reliability with minimized overheads. For example. recent work [30] proposed a low-cost software scheme to improve the reliability of the commodity embedded processors. For the instruction cache in embedded processors, we also need such low-cost solutions to improve its reliability under the power and area constraints.

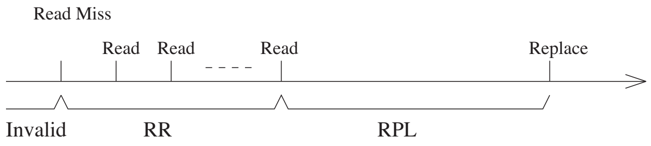

Fig. 1. The lifetime of a data item in the instruction cache.

3. TVF based instruction cache vulnerability characterization

To characterize the vulnerability of the on-chip instruction cache in embedded processors, we first analyze the vulnerability factor by only considering the temporal behavior of the instruction cache itself. We utilize the TVF model proposed in [13]. In this lifetime model, the lifetime of a data item in the instruction cache is divided into three phases according to the previous activity and the current one. They are:

- `RR`: lifetime phase between two consecutive reads of a data item,
- `RPL`: lifetime phase between the last read and the replacement of a data item,
- `Invalid`: lifetime phase when the data item is in the invalid state.

Fig. 1 shows the correlation among three lifetime phases for typical instruction cache activities. These three phases are further categorized into two groups, vulnerable and nonvulnerable. The vulnerable phase is defined by the fact that errors occurring in this phase have the possibility to propagate to the CPU. The `RR` phase is vulnerable phase since error occurring in this phase will propagate to the CPU by the instruction fetch. The `RPL` and `Invalid` are nonvulnerable since the errors occurring during these two phases will be discarded. Note that the data item in the instruction cache can be a cacheline, a 32-bit instruction, or a single bit. It is different from the data item in the data cache where the data item normally refers to a cacheline, a word, or a byte. Since all data items accessed in the instruction cache are of the same size, which is the 32-bit instruction in our simulated ARM embedded processor, we can choose the instruction-based (32-bit) characterization in our TVF study. Although the instruction-based characterization is accurate enough for the vulnerability analysis within the instruction cache, it will become inaccurate for the system-level vulnerability estimation, where we need to track every bit in an instruction for its vulnerability. Therefore, a bit-based characterization will be performed for our SICVF model. The details will be elaborated in the following section.

4. System-level instruction cache vulnerability characterization

The previous section discussed a conventional instruction cache vulnerability characterization model based on the TVF analysis. However, this model only provides an upper bound for the vulnerability factor of the instruction cache [13]. In order to provide more comprehensive understanding on the vulnerability of instruction caches, a system-level vulnerability analysis for instruction caches in embedded processors is needed.