Microelectronics Reliability 54 (2014) 2621–2628

Contents lists available at ScienceDirect

Microelectronics Reliability

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

Evaluating the radiation sensitivity of GPGPU caches: New algorithms and experimental results

D. Sabena [a,*], M. Sonza Reorda [a], L. Sterpone [a], P. Rech [b], L. Carro [b]

[a] Politecnico di Torino, Dipartimento di Automatica e Informatica, Torino, Italy
[b] Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil

ARTICLE INFO

Article history:
Received 1 February 2014
Received in revised form 5 May 2014
Accepted 5 May 2014
Available online 2 June 2014

Keywords:
GPGPUs
Radiation testing
Cache
SEU

ABSTRACT

Given their high computational power, General Purpose Graphics Processing Units (GPGPUs) are increasingly adopted: GPGPUs have begun to be preferred to CPUs for several computationally intensive applications, not necessarily related to computer graphics. However, their sensitivity to radiation still requires to be fully evaluated. In this context, GPGPU data caches and shared memory have a key role since they allow to increase performance by sharing data between the parallel resources of a GPGPU and minimizing the memory accesses overhead. In this paper we present three new algorithms designed to support radiation experiments aimed at evaluating the radiation sensitivity of GPGPU data caches and shared memory. We also report the cross-section and Failure In Time results from neutron testing experiments performed on a commercial-off-the-shelf GPGPU using the proposed algorithms, with particular emphasis on the shared memory and on the L1 and L2 data caches.

1. Introduction

Recently, new devices known as General Purpose Graphics Processing Units (GPGPUs) made their appearance on the market. Their very high computational power, combined with low cost, reduced power consumption, and flexible development platforms are pushing their adoption not only for graphical applications, but also in the High Performance Computing (HPC) market. Moreover, GPGPUs are increasingly used [1] in some safety–critical embedded domains, such as automotive, avionics, space and biomedical. As an example, *Advanced Driver Assistance Systems* (ADASs), which are increasingly common in cars, make an extensive usage of the images (or radar signals) coming from external cameras and sensors to detect possible obstacles requiring the automatic intervention of the breaking system. However, several issues about the reliability of GPGPUs have been raised [2,3]. Given their high degree of parallelism, many assume that GPGPUs could intrinsically provide a good degree of fault tolerance; however, their size and complexity could make them particularly sensible to soft errors. Moreover, while hardening techniques already exist for systems based on traditional CPUs, similar solutions for GPGPU-based systems are still in their infancy [4]. The programming paradigm adopted by GPUs (i.e., Single Instruction Multiple Data) can provide some advantages when designing hardening strategies, but requires effective solutions to combine detection and correction capabilities with the required high performance characteristics.

When assessing the sensitivity of GPGPUs to radiation, a commonly adopted solution is performing radiation experiments with accelerated particles, counting the number of errors they trigger. A major target of radiation effects are GPGPU's memories, both standard and caches. Recently, manufacturers adopted Error Correction Code (ECC) mechanisms against Soft Errors affecting all GPGPU memory modules. In particular, some manufacturers like NVIDIA have recently introduced the ECC scheme for GPGPUs oriented to High Performance Computing (HPC), which is characterized by looser constraints on power consumption and area cost than the embedded computing market. Vice versa, in GPGPUs designed for embedded systems the ECC mechanism is still not available. Radiation errors on memories may then significantly reduce the reliability of the device. Finally (and more generally), few data are still available on the GPGPU memory soft-error rate, so that quantitatively evaluating their reliability is still a hard task.

The purpose of this paper is focused on investigating the sensitivity to soft-errors induced by terrestrial radiation effects on GPGPUs, thus evaluating their capability to produce correct results even when used for long and massive computations in HPC data centers, and to work in harsh environments and/or for safety–crit-

* Corresponding author. Address: DAUIN, Corso Duca degli Abruzzi, 24, 10129, Torino, Italy. Tel.: +39 0110907091; fax: +39 0110907099.
E-mail address: davide.sabena@polito.it (D. Sabena).

http://dx.doi.org/10.1016/j.microrel.2014.05.001
0026-2714/

ical applications. When adopting GPGPUs in these contexts, a major target of radiation effects is represented by the caches, due to their size and their impact in increasing the performance of the GPGPU [5]. Any radiation campaign focused on testing cache soft-error sensitivity requires first forcing the memories to a given value, then exposing the device to a given radiation fluence letting errors to accumulate, and finally checking whether all the bits in the cache(s) are still holding the initial value. The first and last steps are particularly critical, since caches are not directly accessible, and relatively few information is delivered about the cache organization and architecture if commercial-off-the-shelf GPGPUs are considered.

One of the motivations of this work is that, traditionally, the manufacturers of electronic devices perform several tests in order to measure the reliability of their devices; however, for industrial and confidentiality reasons normally this kind of data is not publicly available. Only few manufacturers (e.g., Xilinx) provide the user with actual radiation-induced error rate; to the best of our knowledge, no GPGPU manufacturer does so. Hence, we believe that a method able to evaluate the soft-error sensitiveness of GPGPUs, like the one proposed in our paper, is extremely valuable. Moreover, the GPU producers, like NVIDIA, are highly likely to know the soft-error sensitiveness of the memory array they use, since this data may be provided by the silicon manufacturer. Nevertheless, the operative soft error reliability of the memory array when embedded in the final products may significantly differ from the stand alone memory array. With the proposed method, we are able to evaluate the cross section and the FIT of the main memories used in the NVIDIA GPUs.

In this paper we describe the method to successfully overcome the accessibility issue intrinsic in caches tests, based on specially written programs, which are run immediately before and after irradiating the GPGPU device with a given particles fluence. The proposed approach has also the feature of detecting anomalies in the hit/miss mechanism of data caches caused by cache tag corruptions: while turning a hit into a miss mainly causes performance degradation, the reverse, even if less likely to occur, may have serious effects on results correctness [6]. To the best of our knowledge, this is the first attempt to exploit carefully written programs (such as those proposed in [7]) to support radiation experiments and extract specific information about the embedded memories reliability.

The proposed method has been implemented on a device by NVIDIA with Fermi Architecture and validated in an extensive radiation campaign at the ISIS facility in the Rutherford Appleton Laboratories (RAL) in Didcot, UK. The gathered experimental results demonstrate its effectiveness, and provide interesting data about the sensitivity to radiation of GPGPU devices.

The remainder of the paper is organized as follows: Section 2 overviews some previous works in the area. Section 3 outlines the GPGPU device we addressed and Section 4 introduces the proposed method. Section 5 describes the performed radiation experiments and reports the gathered results. Finally, Section 6 draws some conclusions and outlines our current work.

2. Related work

Radiation effects are a concern for the reliability of electronic devices not only in harsh radiation environments such as space or avionic, but also at ground level. Today, device technology shrinking has led to a drastic reduction of critical charges in logic gates and memory cells that result in a higher sensitivity to soft-errors induced by ionizing radiation. In the last years, an increasing research interest has been devoted to the soft-error sensitiveness evaluation of GPGPUs. A first method for the evaluation of their radiation sensitivity has been proposed in [8], where authors present an analytical model for the evaluation of Single Event Upset (SEU) occurrences on GPGPUs depending on the memory and register usage of the running application. Recently, GPGPUs have been evaluated using spallation neutron sources that provide the user with an atmospheric-like spectrum. A preliminary experimental setup for the execution of neutron radiation test of a GPGPU has been proposed in [9]. The authors describe a low-cost but effective setup providing some guidelines on how to test GPGPUs, focusing on the constraints imposed by the radiation source and the device connections with the host computer controlling the experiment. The first radiation test data demonstrate that both memory and logic resources of a GPGPU may be corrupted by atmospheric neutrons. Being characterized by high-performance computational units such as fixed and floating point units, a further evaluation of the probabilities that radiation-induced errors may affect the mantissa, the exponents or the sign has been evaluated in [10]. A strong variation on the output error rate has been observed when different types of data are elaborated showing a higher sensitivity for the resources used by the mantissa. Moreover, radiation evaluations also addressed the distribution of the computational workload on GPGPUs such as the thread distribution and their relation to multiple output errors [11].

Aside from testing approaches, software-based hardening solutions have also tentatively addressed in [12] where authors developed an optimized software-based hardening strategy exclusively oriented to matrix multiplication applications. Researches have also investigated the possibility of using Software-Based Self-Test (SBST) programs in order to detect and localize permanent faults in GPGPUs: a preliminary work proposing a possible SBST solution has been presented in [13]. Actually, none of the previous developed works addressed the measurement of the sensitivity of the memory areas of a GPGPU, which involves shared memories and different caches levels. The present work goes in that direction. We provide a set of methods based on monitoring programs capable to initialize and observe the soft-error effects on the different memory levels belonging to GPGPUs. The main scientific contribution provided by our method is the possibility of effectively evaluating the impact of soft-errors occurring into the arrays and control circuitry of GPGPUs cache memories through monitoring programs. This allows evaluating the expected soft-error rate into memory cells belonging to shared memories and caches of L1 and L2 levels, independently from the running application and without any intrusiveness into the radiation test data obtained. Furthermore, our method also offers the possibility of counting the number of erroneous phenomena affecting the L1 and L2 cache index logic resources.

3. GPGPUs internal structure

3.1. NVIDIA fermi architecture

The GPGPU Architecture we address in this paper is the NVIDIA Fermi Architecture. With over three billion transistors and featuring up to 512 CUDA cores, the Fermi Architecture delivers supercomputing features and performance at 1/10th of the cost and 1/20th of the power of traditional CPU-only servers [14]. The Fermi GPU family is built around an array of Streaming Multiprocessors (SMs), as shown in Fig. 1, each of which has the ability of executing several threads in parallel [12]. The SMs are composed of several (typically from 32 up to 48) computational units, called NVIDIA CUDA cores or Streaming Processors (SPs), as shown in Fig. 2, where each core manages a thread at a time. From the software point of view, CUDA extends C language by allowing the programmer to define C-based functions, called *kernels*, that, when called, are executed in parallel by N different CUDA threads; the NVIDIA