

Microarchitecture level reliability comparison of modern GPU designs: First findings

*Original*

Microarchitecture level reliability comparison of modern GPU designs: First findings / Vallero, Alessandro; DI CARLO, Stefano; Tselonis, Sotiris; Gizopoulos, Dimitris. - STAMPA. - (2017), pp. 129-130. (Intervento presentato al convegno 2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2017) tenutosi a USA nel 24-25 April 2017) [10.1109/ISPASS.2017.7975280].

*Availability:*

This version is available at: 11583/2678586 since: 2017-09-08T18:25:54Z

*Publisher:*

Institute of Electrical and Electronics Engineers Inc.

*Published*

DOI:10.1109/ISPASS.2017.7975280

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

IEEE postprint/Author's Accepted Manuscript

©2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

# Microarchitecture Level Reliability Comparison of Modern GPU Designs: First Findings

Alessandro Vallero, Stefano Di Carlo  
Politecnico di Torino, Italy  
{stefano.dicarlo,alessandro.vallero}@polito.it

Sotiris Tselonis, Dimitris Gizopoulos  
University of Athens, Greece  
{dgizop, tseloniss }@di.uoa.gr

**Abstract** – State-of-the-art GPU chips are designed to deliver extreme throughput for graphics as well as for data-parallel general purpose computing workloads (GPGPU computing). Unlike graphics computing, GPGPU computing requires highly reliable operation. The performance-oriented design of GPUs requires to jointly evaluate the vulnerability of GPU workloads to soft-errors with the performance of GPU chips.

We briefly present a summary of the findings of an extensive study aiming at the evaluation of the reliability of four GPU architectures and corresponding chips, orrelating them with the performance of the workloads.

**Keywords** – GPGPU, microarchitecture, simulator, reliability, performance, fault injection, throughput

## I. INTRODUCTION

Recently, the research community has started tackling the challenging problem of characterizing the reliability of GPGPU based systems, i.e., their vulnerability to soft and hard errors [1] [2]. This challenging problem requires the development of accurate and fast reliability assessment techniques to deal with the delicate trade-off between analysis time and accuracy of the reported measurements and the ability to provide results that can guide system designers in the choice and development of efficient error resilience mechanisms. The main goal of this paper is to show some findings of an extensive study aimed at evaluating the hardware and software features that influence the reliability of GPGPU chips in the presence of soft-errors. Among the different structures composing GPU architectures we focus on the register file and on the local (AMD terminology) or shared (NVIDIA terminology) memory. The full scale of the study considers several important aspects including correlation between reliability and performance, resource sizes, resource occupancy and the execution scheduling. Different reliability assessment methodologies are employed to identify trade-offs between analysis time and accuracy of results. GPUs from different vendors, architectures and programming models are compared: AMD Southern Islands, NVIDIA G80, GT200 and Fermi. Reliability of all devices is analyzed running the same set of 10 benchmarks, written using the corresponding language: OpenCL for AMD GPUs and CUDA for NVIDIA GPUs. To our knowledge, this is the first work comparing the reliability and correlating it to the performance for the most important GPU families of different vendors, microarchitectures, Instruction Set Architectures (ISAs) and computational models using the same set of benchmarks and employing the most prominent evaluation methodologies used in reliability evaluation.

## II. GPUS RELIABILITY EVALUATION FRAMEWORK

This study has been carried out using two tools named GUFU and SIFI<sup>1</sup>. GUFU, previously presented in [4] has been developed to perform reliability analysis using fault injection and ACE-based analysis on NVIDIA GPUs. It is based on the GPGPU-Sim [6]. Similarly to GUFU, SIFI is a new fault injection and ACE analysis tool developed to characterize AMD GPUs. SIFI is built on top of the Mult2Sim micro architectural simulator and models the Southern Island family of AMD GPUs [3]. For both tools, the reliability analysis has been always performed considering the low-level assembly code that represents the binary code running on the real hardware. For this reason, for NVIDIA GPUs, the SASS assembly was preferred to PTX. This allowed a fair comparison of NVIDIA and AMD architectures by injecting faults on the actual hardware registers for both GPU families. The brief findings presented in this paper compare the Architectural Vulnerability Factor (AVF) of the hardware structures for different GPUs using both ACE analysis and fault injection. The AVF quantifies the probability that a bit-flip (soft-error) affecting a hardware structure will manifest as an error at the system output. The AVF is a pure reliability metric and does not provide a fair comparison among GPUs with different clock frequencies, instruction sets and microarchitectures. A system designer can be provided with a broader idea of the system performance and reliability for any given workload when combined metrics are used. Such a metric can be the rate of Executions per Failure (EPF). EPF can be defined as the number of complete executions of a benchmark between failures and depends on all parameters that affect both performance and reliability (clock frequency, ISA, microarchitecture, AVF, components size, program execution time, etc.). We define EPF as the ration between the executions in time (EIT), i.e., the number of executions of a benchmark in  $10^9$  hours of device operation and the failures in time  $FIT_{GPU}$ , i.e., the number of failures in  $10^9$  hours of device operation ( $EPF = EIT / FIT_{GPU}$ ). A similar metric to correlate reliability (failures in time) and performance (executions in time) is also used for a CPUs vs. GPUs comparison in [7].

## III. RELIABILITY EVALUATION

For our evaluation, we used 10 benchmarks: 7 available both in the CUDA SDK2 and AMD-APP SDK3 and 3 from

<sup>1</sup> GUFU is based on GPGPU-Sim-3.2.2 while SIFI is based on Multi2Sim-4.2.

<sup>2</sup> <https://developer.nvidia.com/cuda-toolkit-42-archive>

<sup>3</sup> <http://developer.amd.com/tools-and-sdks/opencl-zone/amd-accelerated-parallel-processing-app-sdk/>

Rodinia benchmarks suite [5]. For every benchmark both the CUDA and the OpenCL implementation is available.

We start the analysis of the reliability of the different architectures and benchmarks by looking at the AVF measurements summarized in Fig. 1 for the vector register file and Fig. 2 for the local memory and computed using both Statistical Fault Injection (FI)<sup>4</sup> and ACE Analysis (ACE). Results show that the AVF can have significant variations moving from one application to another but also variations can be observed for the same application executed on different GPUs. This confirms the need of carefully performing this type of analysis. This is particularly true for the local memory in which a clear trend in the way the AVF changes between different GPU architectures cannot be identified, suggesting that this analysis must be carefully executed on a case by case. Red lines reporting the occupancy of the considered memory structures show a strong correlation of the AVF with this parameter. It is interesting to note that while for the register file the ACE analysis significantly overestimates vulnerability compared to FI, the same technique is very accurate (very close to FI) for the local memory, suggesting that for this structure ACE analysis can be used without significant loss of accuracy but with significant gain in the required simulation time compared to long FI campaigns.

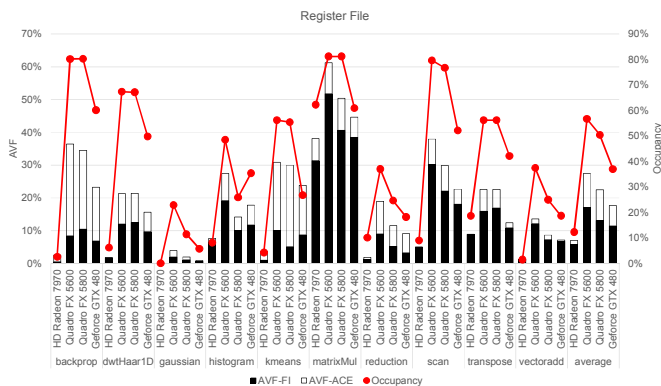


Fig. 1. AVF for Register File measured by fault injection (FI) and ACE.

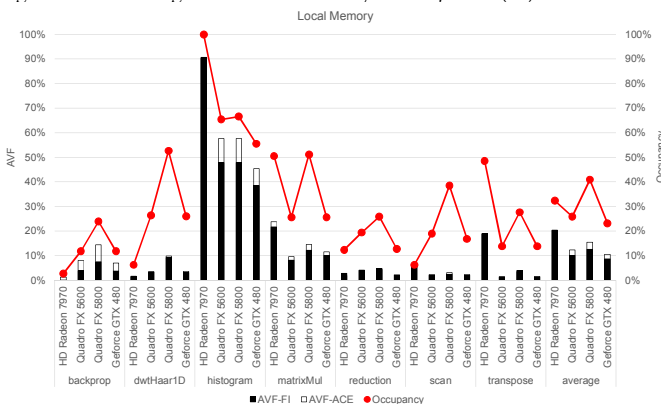


Fig. 2. AVF for Local Memory measured by fault injection (FI) and ACE.

Fig. 3 shows the EPF for the considered GPU models, intuitively reporting the throughput of a machine in finalizing

correct program executions per failure. The EPF metric is useful to the architects who can quantify the effectiveness of a hardware based error protection technique, which can be applied to their designs (if needed) along with a performance cost. Larger EPF numbers show a larger number of executions between failures and different protection mechanisms can deliver different improvements in the FIT rates and can also have different impact on performance. Combining performance and reliability measurements in the EPF metric delivers a broader view for decision-making.

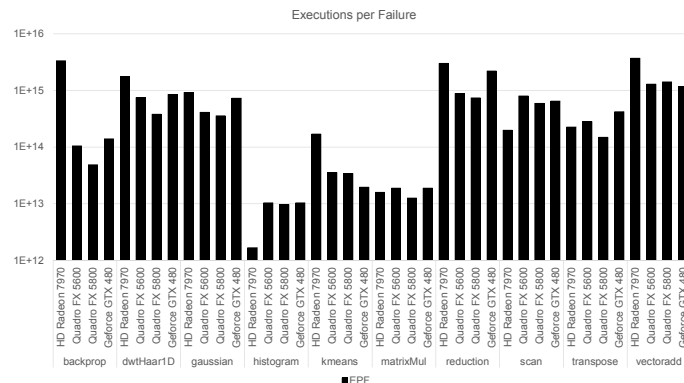


Fig. 3. Executions per Failure (EPF)

#### IV. CONCLUSIONS

We have presented a first summary of our findings in comparing reliability metrics for different state-of-the-art AMD and NVIDIA GPUs. Our reliability measurements (AVF and EPF) are computed using both FI and ACE analysis to reveal the differences between the two approaches and to show the value of being able to perform this type of analysis when designing a GPGPU system. A larger set of experiments and results is underway and will be available in future publications.

#### ACKNOWLEDGMENT

This work was funded by the European Union through the CLERECO FP7 Project (Grant Agreement 611404).

#### REFERENCES

- [1] M.Riera, R.Canal, J.Abella, A.Gonzalez, "A Detailed Methodology to Compute Soft Error Rates in Advanced Technologies," DATE 2016.
- [2] P.Rech, L.L.Pilla, P.O.A.Navaux, L.Carro, "Impact of gpus parallelism management on safety-critical and hpc applications reliability," DSN 2014.
- [3] R.Ubal, B.Jang, P.Mistry, D.Schaa, D.Kaeli, "Multi2Sim: a simulation framework for CPU-GPU computing," PACT 2012.
- [4] S.Tselonis, D.Gizopoulos, "GUF: a Framework for GPUs Reliability Assessment," ISPASS 2016.
- [5] S.Che, M.Boyer, J.Meng, D.Tarjan, J.W.Sheaffer, S.-H.Lee, K.Skadron, "Rodinia: A benchmark suite for heterogeneous computing," IISWC 2009.
- [6] A.Bakhoda, G.Yuan, W.W.L.Fung, H.Wong, T.M.Aamodt, "Analyzing CUDA Workloads Using a Detailed GPU Simulator", ISPASS 2009.
- [7] A.Chatzidimitriou, M.Kaliorakis, S.Tselonis, D.Gizopoulos, "Performance-Aware Reliability Assessment of Heterogeneous Chips", VTS 2017.

<sup>4</sup> We simulated 2,000 fault injections per hardware structure, which statistically provides 2.88% error margin for 99% confidence level.