2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) | 978-1-7281-8643-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPASS51385.2021.00040

The Impact of SoC Integration and OS Deployment on the Reliability of Arm Processors

Pablo Bodmann* George Papadimitriou† Dimitris Gizopoulos† Paolo Rech△
*PPGC, Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brasil
prjbodmann@inf.ufrgs.br
†Dept. of Informatics and Telecommunications, University of Athens, Athens, Greece
{georgepap | dgizop}@di.uoa.gr
△Politecnico di Torino, Turin, Italy
paolo.rech@polito.it

***Abstract*—Arm CPU architectures, thanks to their efficiency and flexibility, have been widely adopted in portable user devices such as smartphones, tablets, and laptops. Recently, the high computing efficiency, together with the unique possibility that Arm offers to adapt the architecture for a specific application, pushed the adoption of Arm-based systems both in HPC (High Performance Computing) applications and autonomous vehicles. The possibility of modifying Arm architecture can potentially be extremely beneficial, as selective fault tolerance solutions can be added at the microarchitectural level. The current trend in the design of computing devices is to integrate several functionalities on the same SoC (System-On-Chip). Modern SoCs usually integrate (one or more) CPUs and (one or more) accelerators, such as GPUs (Graphics Processing Units) or FPGAs (Field Programmable Gate Arrays). These SoCs typically allow the computing cores to share common memories, which significantly improves the performance and reduces the total power consumption but may impact the system's reliability. In this work, we have evaluated the impact of SoC integration and OS deployment using beam experiments and microarchitectural fault injection.**

I. Methodology and Goals

In this work we consider the reliability of Arm Cortex-A5 and Cortex-A9 CPUs with and without an Operating System (Linux). We use a dual reliability evaluation setup. We combine the realistic physical measurement of the FIT (Failures-in-Time, i.e., failures per 10^9 hours of operation) rates obtained through extensive beam experiments on actual chips (equivalent to 18 million years of natural exposure) with the fine-grain and detailed analysis of faults propagation obtained with microarchitecture fault injection in early-stage CPU models (based on 176,000 injections). The comprehensive comparison of microarchitecture level fault injection with beam experiments is a step forward in the quest for early FIT rate predictions because the microarchitectural description is available much earlier than the silicon prototype, or the final (Commercial off-the-shelf) COTS device. The Silicon A5 is implemented in a 65nm CMOS technology in the Microchip SAMA5D2 XPLAINED ULTRA board and a Cortex-A9 implemented in a Xilinx Zynq™-7000 SoC which uses a 28nm CMOS technology. Since the Cortex-A5 and Cortex-A9 are implemented in two different technologies, the raw probability for a neutron to generate a fault is different in the two devices. The experimentally measured FIT rates for a single memory bit in the L1 cache are 2.37×10^{-4} for the Cortex-A5 and 2.59×10^{-5} for the Cortex-A9.

The main contributions of this work are: an evaluation of the impact of SoC integration and OS deployment in Arm CPUs reliability based on beam experiments; a fine-grained microarchitectural component-based fault injection analysis of the vulnerability of codes executed on top of Linux and bare-metal in Arm CPUs; and hints and guidelines on how to estimate, pre-silicon, the reliability of a CPU as stand-alone or integrated on a SoC and when executing programs on top of an OS.

II. Related Works

Although the reliability of standalone CPUs, FPGAs, and GPUs has already been extensively studied [1]–[7], there are few works that evaluate the impact of SoC integration on a CPU core reliability. Moreover, previous works have evaluated the effects of the OS on code reliability and even designed radiation-hardened OSes [8]–[11], however, it is still unclear which are the architectural vulnerabilities that induce errors in an OS.

III. Results

A. Beam Experiments

On our beam experiments, we observed that *the average SDC rates are very similar for the two devices* (the average SDC rate difference between the A5 and A9 is 12% for bare metal and 45% for Linux). Interestingly, *the average DUE rate (Crash, AppCrash, SysCrash) is at least one order of magnitude higher on the A9*. Correlating the technology sensitivity data (the A5 has a 9.1x higher sensitivity than the A9, but A5 and A9 have similar SDC rates, and the A9 has a 10x higher DUE rate) suggests that the A9 architecture is more vulnerable than the A5, mostly for DUEs. Additionally, as the A9 is bigger (out-of-order and with a four times larger L2 cache) than the A5, it is likely to have a higher probability of being hit by a neutron.

Moreover, when comparing results obtained running the codes bare-metal and on top of Linux we found out that *The OS presence only slightly increases the SDC FIT rate*, of about 2.4x for the A5 and 1.9x for the A9, on average. While, at a first glance, the OS impact seems high, it is much lower than the differences between SDC FIT rates of different codes in the same device (and of the impact on FIT rates of other factors, as shown in [12], [13]).

For DUEs, the bare-metal execution can only lead to a Crash (i.e. the application and, then, the device are not responding), while the execution on top of Linux can lead to AppCrash (i.e., the application does not respond but Linux is still up and running) and SysCrash (i.e., Linux not responding). When an OS is used the overall DUE rate (AppCrash + SysCrash) increases by 8.5x for the A5 and 2.6x for the A9, on the average.

Crashes in bare-metal are mainly caused by exceptions raised by the CPU and AppCrashes on Linux are results from the kernel terminating the application. The reason for the kernel terminating the application may be the result of a CPU exception handled by the kernel, but also from a signal sent by the kernel to the application. From our experiments, we can observe that *for both the A5 and A9, the average AppCrash is about 50% higher than the bare-metal*

978-1-7281-8643-6/21/$31.00 ©2021 IEEE
DOI 10.1109/ISPASS51385.2021.00040

crash. The SysCrash, on the contrary, is triggered by the corruption of kernel code or data, interfaces, etc. An error in the application being executed can hardly affect the system since it operates in an unprivileged space. The reported experimental data shows that *the SysCrash rate is almost constant for both the A5 and on the A9* (the variation between codes is, on average, 30%).

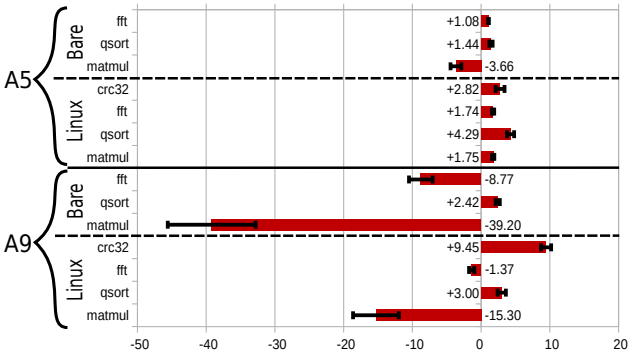

Fig. 1. Beam and fault injection SDC FIT rates comparison

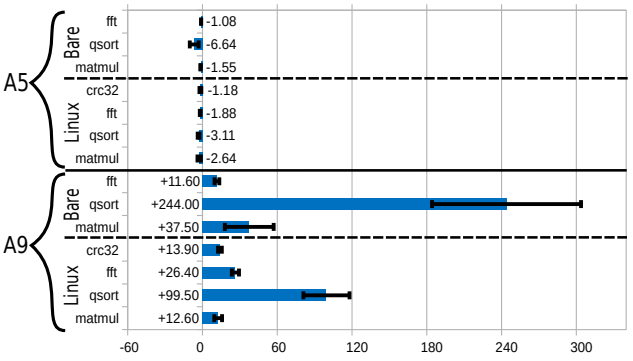

Fig. 2. Beam and fault injection Crash/AppCrash FIT rates comparison

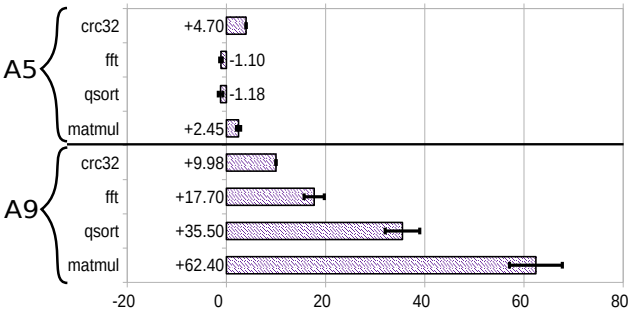

Fig. 3. Beam and fault injection SysCrash FIT rates comparison

B. GeFIN and Beam Comparison

Another important contribution of this work is the evaluation of the accuracy of microarchitectural fault injection in estimating the FIT rate of codes executed in silicon devices. This comparison also helps to see the effects of the OS stack and the SoC integration. Figure 1, 2, and 3 show the comparison between FIT rates measured with beam experiments and predicted with GeFIN (an accurate microarchitecture-level reliability assessment framework [14]) for SDC, Crashes/AppCrashes, and SysCrashes, respectively. The results shown on all figures are done by dividing, for each code, the highest FIT rate between the one calculated with beam and the one predicted using fault injection by the lowest FIT rate between the two. Whenever the FIT rate obtained with beam experiments is higher than the fault injection, the value is represented as positive; negative otherwise. That is, in Figure 1, for matmul executed in bare-metal on the A5, the SDC FIT rate predicted with fault injection is 3.66x higher than the one measured with beam experiments.

A result of great impact, shown in Figure 1, is that for the majority of cases, the GeFIN SDC FIT rate prediction is very similar to the one measured with beam experiments (smaller than 5x) on both boards on almost all benchmarks but matmul and FFT on the A9. This result attests that *microarchitectural fault injection can be used to predict the SDC FIT rate* of the correspondent real hardware of two completely different devices, with and without Linux.

When estimating Crashes and AppCrashes, shown in Figure 2, for the A5 GeFIN reports FIT rates very similar to the ones measured with beam experiments, for both bare-metal and Linux configurations. Even for SysCrashes, the FIT rates for the A5 are very similar between fault injection and beam experiments. As observed for SDC, then, *for stand-alone CPUs GeFIN accurately models DUEs (Crashes, AppCrashes, and SysCrashes), considering also the impact of the OS*.

For the A9 the differences diverge, with beam providing a Crash and AppCrash FIT rate from 1 to 2 orders of magnitude higher than GeFIN, for both Linux and bare-metal. We can derive that *a great portion of DUEs in the embedded A9 are generated by faults in resources that are not modeled in gem5*. For the single-core, stand-alone A5, the model used on gem5 and the real hardware are very similar. This is not the case with the board with the A9, which is integrated with a FPGA (not used) and is dual-core (with one core disabled).. The integration in a SoC requires additional interfaces, logical or control components that are not modeled in gem5. This highly complex interconnect inside the chip causes a DUE when corrupted.

IV. Conclusions

In conclusion, we have performed an extensive reliability evaluation of two widely used Arm microprocessors, Cortex-A5 and Cortex-A9, with the primary objective of identifying the impact that the SoC integration and the OS deployment have on the FIT rates. The two cores are fabricated using different technologies and different SoC organizations: A5 is an individual CPU core whereas the A9 is integrated in a SoC. Our analysis relies on a unique combination of neutron beam testing and microarchitecture-level fault injection to understand the contribution of the different aspects: the bare CPU, the SoC integration, and the OS deployment. The primary finding of our analysis is that the SDC FIT rate is only slightly affected by the SoC integration and the presence of the OS. On the other hand, DUEs (both application and system crashes) are dramatically increased (up to 2 orders of magnitude) due to SoC integration.

References

[1] G. Wang, S. Liu, and J. Sun, "A dynamic partial reconfigurable system with combined task allocation method to improve the reliability of fpga," *Microelectronics Reliability*, vol. 83, pp. 14 – 24, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026271418300350

[2] A. Vallero, A. Carelli, and S. Di Carlo, "Trading-off reliability and performance in fpga-based reconfigurable heterogeneous systems," in *2018 13th International Conference on Design Technology of Integrated Systems In Nanoscale Era (DTIS)*, April 2018, pp. 1–6.
[3] A. Chatzidimitriou, P. Bodmann, G. Papadimitriou, D. Gizopoulos, and P. Rech, "Demystifying soft error assessment strategies on arm cpus: Microarchitectural fault injection vs. neutron beam experiments," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2019, pp. 26–38.
[4] M. Wilkening, F. Previlon, D. R. Kaeli, S. Gurumurthi, S. Raasch, and V. Sridharan, "Evaluating the resilience of parallel applications," in *2018 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Oct 2018, pp. 1–6.
[5] A. Vallero, S. Tselonis, D. Gizopoulos, and S. Di Carlo, "Multi-faceted microarchitecture level reliability characterization for nvidia and amd gpus," in *2018 IEEE 36th VLSI Test Symposium (VTS)*, April 2018, pp. 1–6.
[6] F. F. d. Santos, P. F. Pimenta, C. Lunardi, L. Draghetti, L. Carro, D. Kaeli, and P. Rech, "Analyzing and increasing the reliability of convolutional neural networks on gpus," *IEEE Transactions on Reliability*, vol. 68, no. 2, pp. 663–677, June 2019.
[7] D. Oliveira, L. Pilla, M. Hanzich, V. Fratin, F. Fernandes, C. Lunardi, J. Cela, P. Navaux, L. Carro, and P. Rech, "Radiation-Induced Error Criticality in Modern HPC Parallel Accelerators," in *Proceedings of 21st IEEE Symp. on High Performance Computer Architecture (HPCA)*. ACM, 2017.
[8] R. K. Iyer and D. J. Rossetti, "Effect of system workload on operating system reliability: A study on ibm 3081," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, pp. 1438–1448, Dec 1985.
[9] T. Santini, P. Rech, L. Carro, and F. R. Wagner, "Exploiting cache conflicts to reduce radiation sensitivity of operating systems on embedded systems," in *2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, Oct 2015, pp. 49–58.
[10] T. Santini, L. Carro, F. R. Wagner, and P. Rech, "Reliability analysis of operating systems and software stack for embedded systems," *IEEE Transactions on Nuclear Science*, vol. 63, no. 4, pp. 2225–2232, Aug 2016.
[11] T. Santini, C. Borchert, C. Dietrich, H. Schirmeier, M. Hoffmann, O. Spinczyk, D. Lohmann, F. R. Wagner, and P. Rech, "Effectiveness of software-based hardening for radiation-induced soft errors in real-time operating systems," in *Architecture of Computing Systems - ARCS 2017*, J. Knoop, W. Karl, M. Schulz, K. Inoue, and T. Pionteck, Eds. Cham: Springer International Publishing, 2017, pp. 3–15.
[12] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *Device and Materials Reliability, IEEE Transactions on*, vol. 5, no. 3, pp. 305–316, Sept 2005.
[13] V. Fratin, D. Oliveira, C. Lunardi, F. Santos, G. Rodrigues, and P. Rech, "Code-dependent and architecture-dependent reliability behaviors," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2018, pp. 13–26.
[14] A. Chatzidimitriou and D. Gizopoulos, "Anatomy of microarchitecture-level reliability assessment: Throughput and accuracy," in *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, Apr 2016. [Online]. Available: https://doi.org/10.1109/ispass.2016.7482075