Comprehensive Reliability Evaluation for Dependable Embedded Systems

Yohan Ko
Yonsei University, Seoul, Korea
yohan.ko@yonsei.ac.kr
Estimated graduation date: February 2018
Advisor: Kyoungwoo Lee (Associate Professor)

1. MOTIVATION

When we consider a wide range of embedded systems, it is essential to consider multiple parameters, such as performance, power, and even reliability [8]. A low power design is just as important as high performance since mobile embedded systems run on limited capacity batteries with a small form factor. In order to meet both requirements, supply voltage is lowered through the aggressive technology scaling. However, decreasing the supply voltage only increases the vulnerability of the systems due to soft errors, which are transient faults induced mainly by energetic particles such as neutrons, protons, and even cosmic rays. To make mobile embedded systems resilient against soft errors, several redundancy-based techniques have been presented, but they lead to significant overheads in terms of performance, power consumption, and hardware area [3, 6]. Selective protections have been presented as alternative for cost-effective protections [7], but how can we ensure whether it is effective or not? We can estimate overheads in terms of runtime, energy, and area, but it is challenging to estimate reliability in a quantitative manner.

Several frameworks have been proposed in order to estimate the reliability for microarchitectural components in a processor [2]. However, there is still a necessity for a validated, comprehensive, and flexible vulnerability estimation modeling. First off, previous works are incomprehensive as they evaluate just a subset of the microarchitectural components in a processor. Secondly, most of the previous works cannot provide configurable vulnerability modeling (e.g., varying ISA, changing number of sequential elements in components) and accurate vulnerability modeling which is due to the underlying simulation platform used. Thirdly, the vulnerability modeling in previous tools is not accurate due to coarse-grained modeling, and its accuracy has not been validated through extensive fault injection campaigns or other schemes. Lastly, many previous tools do not accurately and quantitatively estimate the reliability in the presence of protection schemes.

In order to perform early design space explorations, we have estimated the reliability of microarchitectural components in a processor based on cycle-accurate gem5 simulator [1]. If we can estimate reliability in a quantitative manner, it enables us to answer key design questions such as: (i) Can hardware architects improve the vulnerability by just configuring hardware options with comparable performance overheads? (ii) Can software engineers improve the hardware-level reliability against soft errors? (iii) System designers can alternate ISAs, but how can they know that protection mechanisms for the previous ISA still works for alternative ISA? Further, our framework can also estimate the reliability with considering protection techniques since several protection techniques already have been implemented even for commodity embedded processors.

2. OUR APPROACH

In order to estimate reliability of processors, we have implemented gemV, the first accurate and comprehensive vulnerability estimation toolset, as shown in Figure 1. Our gemV is configurable and extensible to analyze future/novel architecture and microarchitecture designs [9]. Our framework provides both vulnerability and runtime at the early design phase based on open-source software simulator, gem5 [1]. This enables us to increase the reliability in a variety of applications without implementing a physical hardware prototype. Therefore, the vulnerability modeling framework can be utilized in a diversity of both industry and academia. In our demonstrations of the capabilities of gemV, we have performed a wide range of design space explorations.

For hardware designers, our gemV provides optimal hardware configurations in terms of performance and reliability. We observe that vulnerability varies by changing architectural parameters like the number of entries in reorder buffer (ROB), instruction queue (IQ), load store queue (LSQ), and pipeline queues. Among configurations, there is an interesting design configuration with 82% less vulnerability at most 1% performance penalty without any protection techniques for the same application. Thus, our gemV framework provide optimal hardware configurations in terms of reliability and performance before prototyping read hardware architectures.

Software designers can use our gemV to find the least vulnerable algorithm for a program. Assume that a software designer wants to sort an integer array. We consider an array sorting application with five sorting algorithms (bubble, quick, insertion, selection, and heap sorting), two compilers (GCC and LLVM), and four optimization levels (no optimization, O1, O2, and O3). For this integer array sorting

DAC '17 Austin, Texas USA
© 2017 ACM. ISBN 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

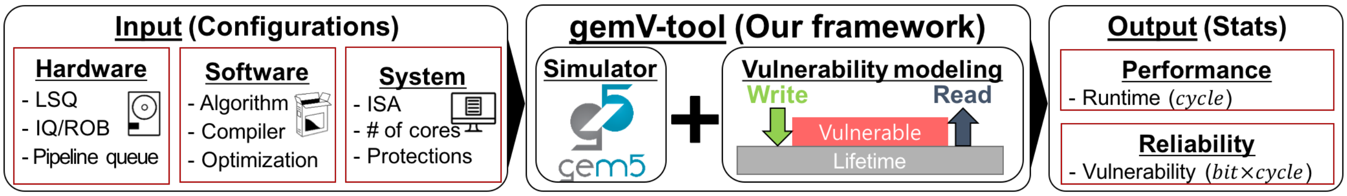

Figure 1: Diagram of our gemV-tool

application, switching from a selection sort algorithm at O1 level of optimization to quicksort at O3 level of optimization can reduce runtime by 53% and vulnerability by 91%. Thus, software engineer can improve the hardware reliability without any hardware modification with our gemV framework.

With the perspective of system designers, it is interesting that the distribution of vulnerabilities among microarchitectural components is sensitive to the ISA. While protecting register rename map and register file will be the most effective in SPARC architecture (more than 75% vulnerability reduction), but the same protection will only reduce the vulnerability by 21% in ARM architecture. In contrast, protecting history buffer and IQ will be the most effective in ARM architecture in our study. Thus, system designers can alternate ISAs for cost-effective protections by using our gemV framework.

We have also estimated vulnerability of parity-protected caches in order to formulate guidelines for the design of power-efficient and reliable L1 data caches [4, 5]. We have found several counter-intuitive and interesting results from our gemV framework. First off, checking parity at reads only (and not at writes) provides 11% more protection with 30% lesser power overheads as compared to that at both reads and writes. Secondly, implementing parity at the word-level granularity provides 53% improved protection as compared to block-level parity implementation. Dirty-bits in the cache should also be implemented at the same granularity, otherwise, there is no improvement in protection. We have found that several popular commercial processors even the ones specifically designed for reliability do not follow these design guidelines from our gemV, and it can result in sub-optimial designs in terms of reliability and performance.

3. SUMMARY

Reliability is one of the most important design concerns in modern embedded systems as technology advances. However, power consumption and performance also should be considered for real-time and resource-constrained embedded systems. For embedded systems, energy consumption can be estimated by energy model or additional power measurement tools, and the runtime also can be easily computed by measuring ticks. On the other hand, reliability is not easy to be estimated even though we need to estimate the reliability in a quantitative manner in order to resolve the complexity of trade-offs among multi-dimensional parameters.

Therefore, this thesis proposes a gemV, which is a comprehensive, accurate, validated, and protection-aware vulnerability estimation toolset, based on cycle-accurate system-level simulator in order to explore design space in terms of performance and reliability. Our gemV framework can provide vulnerability-aware hardware configurations, software development, and system design at the early design phase. Our future work will include in-depth design space exploration from various perspectives such as profiling-guided optimizations, programming guidelines or protection schemes.

4. REFERENCES

[1] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 39(2):1–7, Aug. 2011.

[2] X. Fu, T. Li, and J. Fortes. Sim-SODA: A unified framework for architectural level software reliability analysis. In *International Workshop in Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems*, 2006.

[3] J. Kang, Y. Ko, J. Lee, Y. Kim, H. So, K. Lee, and Y. Paek. Selective validations for efficient protections on coarse-grained reconfigurable architectures. In *IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*, pages 95–98, June 2013.

[4] Y. Ko, R. Jeyapaul, Y. Kim, K. Lee, and A. Shrivastava. Guidelines to design parity protected write-back L1 data cache. In *Design Automation Conference (DAC)*, pages 24:1–24:6, 2015.

[5] Y. Ko, R. Jeyapaul, Y. Kim, K. Lee, and A. Shrivastava. Protecting caches from soft errors: A microarchitect's perspective. *ACM Transactions on Embedded Computing Systems (TECS)*, 2017.

[6] Y. Ko, J. Kang, J. Lee, Y. Kim, J. Kim, H. So, K. Lee, and Y. Paek. Software-based selective validation techniques for robust CGRAs against soft errors. *ACM Transactions on Embedded Computing Systems (TECS)*, 15(1):20:1–20:26, 2016.

[7] J. Lee, Y. Ko, K. Lee, J. M. Youn, and Y. Paek. Dynamic code duplication with vulnerability awareness for soft error detection on VLIW architectures. *ACM Transactions on Architecture and Code Optimization (TACO)*, 9(4):48:1–48:24, 2013.

[8] M. Shafique, S. Garg, J. Henkel, and D. Marculescu. The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In *Design Automation Conference (DAC)*, pages 185:1–185:6, 2014.

[9] K. Tanikella, Y. Koy, R. Jeyapaul, K. Lee, and A. Shrivastava. gemV: A validated toolset for the early exploration of system reliability. In *International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 159–163, July 2016.