Evaluating Soft Error Reliability in Multi-core Processors using OVPSim-FIM

Oishi Deb
Software and Electronics Engineering, Undergraduate student
Department of Engineering, University of Leicester, UK
od28@student.le.ac.uk

ABSTRACT

Commercial Off-The-Shelf (COTS) processors have been widely used in embedded systems. The increasing complexity of underlying processors along with the increasing chip power densities and the continuous technology shrink is making COTS-based systems more vulnerable to soft errors. The soft errors occurrence may cause critical failures on system behavior. This project utilised a fault injection framework, called OVPSim-FIM to access the soft error reliability of different ARM instruction set architectures and evaluates the finding.

KEYWORDS

Multi-core systems, Soft errors, OVPSim-FIM, Fault Injection.

1 INTRODUCTION

In today's age of technology, the demand for high performance in multi-core systems is always increasing. The processors in these systems may need to operate under intense clock frequencies and various voltage domain to achieve high performance, which makes these vulnerable to soft errors. A soft error is a popular reliability issue in an embedded system, which is caused by electromagnetic radiation or ions striking a sensitive node in a semiconductor memory, microprocessor or a transistor. As the embedded system is becoming such integral part of our life, so the motivation of this project is to investigate how virtual platforms can be used to detect soft errors in the early design phase for these multi-core processors. This project uses the Open Virtual Platform Simulator (OVPSim) Fault Injection Module (FIM) to analyse soft error reliability.

2 EXPERIMENTAL DETAILS

2.1 OVPSim-FIM

Using the tool OVPSim-FIM, single-bit Fault Injection (FI) was performed in 32-bit ARMv7 and 64-bit ARMv8 instruction set architectures, considering multicore processors executing Linux Kernel and Rodinia benchmarks with up to 10 million of object code instructions. The 800 faults were injected in the General Purpose Registers (GPA). The result of these faults was then classified according to Cho et al. [1] which are – Vanished, Output Not Affected (ONA), Output Mismatch (OMM), Unexpected Termination (UT) and Hang.

3 RESULTS AND DISCUSSION

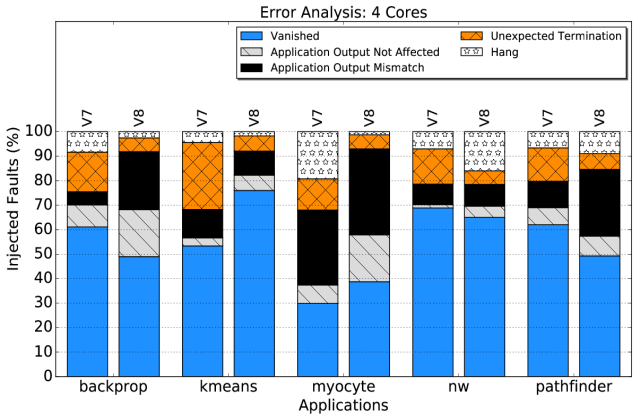

Figure 1: Comparison between V7 and V8 architecture when injected with 800 faults for five Rodinia benchmark applications.

Fig.1 shows the fault injection result for V7 and V8. The errors which cause a different number of executed cycles for the applications but with the memory and architectural state preserved were considered to be ONA. While the errors which caused the applications to have a different number of executed cycles, corrupted memory states, and architectural state were considered to be OMM. Out of the five error classifications having OMM is very critical however ONA is not a problem as the memory is not affected. Vanished means that the trace of error was gone after FI, hence having a higher percentage of vanished can be considered as more reliable. From Fig.1, it is clear that the pattern of the fault occurred is dependent on the application. The ARMv8 protects more memory structures with the Error-Correcting Code (ECC) as compare to ARMv7.

4 CONCLUSIONS

In summary, collected faults were successfully analysed according to different criteria and it has been observed that the percentage of vanished faults have been higher in V8 than V7 for certain applications. The author concludes that V8 is more reliable architecture in certain scenarios depending on the specific purpose. Obtaining these data using the simulator, will help to minimize the soft error occurrence in the early design phase of processors and hence saving life and financial losses in the real world.

REFERENCES

[1] H. Cho, et al. "Quantitative evaluation of soft error injection techniques for robust system design," in 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), 2013, pp. 1–10.