Week 1 Journal, CS6000

Mong Sim
Computer Science Department
University of Colocrado Colorado Springs
Colorado Springs, USA
msim@uccs.edu

***Abstract*—I am a second year Ph.D. student 1 in the Computer Science department, and my research areas are in Computer Architecture and Compiler. Both my undergraduate and graduate studies were in Electronic and Electrical Engineering. I am taking my Ph.D. in the Computer Science department with a firm belief that having a good computer architecture alone is insufficient; we need a good compiler to leverage the innovation. Only the School of Computer Science offers these combinations. I am a Principal Engineer working on the Space Grade Computer. We design and sell Radiation Harden semiconductors to our Aerospace and Military customers. Before coming to this class, I published none paper and hoping at the end of this course I have a complete understanding of what a research paper entails. I am familiar with technical reports and application notes writing. Since the main part of a research paper is to offer a novel idea of something useful to the public at large. I hope to see a guideline showing the sections needed for research writing, and what is the balance between the coverage of the related work and your research. I presume the writing style is academic research writing result, but what is the tone of the writing? Should we gear the design section and the rest of the paper towards conversion or technical report writing style? How are the differences between a Workshop, Conference and Journal paper? To date, I used LaTex once to write a paper. Still, I consider myself as a beginner, but with Google helps, I can get most of the basics done.**

Fig. 1. A Picture of Mong Sim

***Index Terms*—component, formatting, style, styling, insert**

I. About Git

This is my github weblink, https://github.com/monguccs/, Reposities 1, Week-1-Journal. Using the email the Professor provided everything goes well without issue.

II. Five Research Papers

A. Paper 1 - Reducing branch predictor leakage energy by exploiting loops

This paper [1] proposes two cost-effective loop-based strategies to reduce the branch predictor leakage without impacting prediction accuracy or performance. The loop-based approaches exploit the fact that loops usually only contain a small number of instructions and, hence, even fewer branch instructions while taking a significant fraction of the execution time. This paper has 5 citations.

B. Paper 2 - Understanding and Mitigating Covert Channels Through Branch Predictors

This paper [2] analyzes these channels in SMT and single-threaded environments under both clean and noisy conditions. Our results show that the residual state-based channel provides a cleaner signal and is effective even in noisy execution environments with another application sharing the same physical core with the trojan and the spy. This paper has 7 citations.

C. Paper 3 - Branch classification: a new mechanism for improving branch predictor performance

This paper [3] proposes branch classification to help improve the accuracy of branch predictors. Branch classification allows an individual branch instruction to be associated with the branch predictor best suited to predict its direction. This paper has 47 citations. If authors ignore stalls such as cache misses and bus confiicts, the branch penalty is defined as

$$C * ((l - p) * r * ipc)$$

.

D. Paper 4 - Neural methods for dynamic branch prediction

This article presents [4] a new and highly accurate method for branch prediction. The key idea is to use one of the simplest possible neural methods, the perceptron, as an alternative to the commonly used two-bit counters. The source of our predictor's accuracy is its ability to use long history lengths, because the hardware resources for our method scale linearly, rather than exponentially, with the history length. This paper has 41 citations.

E. Paper 5 - A practical low-power memristor-based analog neural branch predictor

This paper [5] demonstrates a practical neural branch predictor based on memristor. By using analog computation techniques, as well as exploiting the accuracy tolerance of branch prediction, our design is able to efficiently realize a neural prediction algorithm. This paper has 1 citation.

REFERENCES

[1] Wei Zhang and Bramha Allu, "Reducing branch predictor leakage energy by exploiting loops,"Journal, ACM Transactions on Embedded Computing Systems (TECS) - SPECIAL ISSUE SCOPES 2005 TECS Homepage archive,Volume 6 Issue 2, May 2007, Article No. 11, ACM New York, NY, USA.

[2] Dmitry Evtyushkin, Dmitry Ponomarev, and Nael Abu-Ghazaleh ,"Understanding and Mitigating Covert Channels Through Branch Predictors," Journa,l ACM Transactions on Architecture and Code Optimization (TACO) TACO Homepage archive, Volume 13 Issue 1, April 2016, Article No. 10, ACM New York, NY, USA.

[3] Po-Yung Chang, Eric Hao, and Tse-Yu Yeh, "Branch classification: a new mechanism for improving branch predictor performance," MICRO 27 Proceedings of the 27th annual international symposium on Microarchitecture, Pages 22-31, San Jose, California, USA — November 30 - December 02, 1994, ACM New York, NY, USA ©1994.

[4] Daniel A. Jiménez and Calvin Lin , "Neural methods for dynamic branch prediction," Journal, ACM Transactions on Computer Systems (TOCS) TOCS Homepage archive, Volume 20 Issue 4, November 2002, Pages 369-397, ACM New York, NY, USA.

[5] Jianxing Wang, Yenni Tim, Weng-Fai Wong, and Hai (Helen) Li, "A practical low-power memristor-based analog neural branch predictor," ISLPED '13 Proceedings of the 2013 International Symposium on Low Power Electronics and Design, Pages 175-180, Beijing, China — September 04 - 06, 2013, IEEE Press Piscataway, NJ, USA ©2013.