

Speculative Return Address Stack Management Revisited

HANS VANDIERENDONCK

Ghent University

and

ANDRÉ SEZNEC

IRISA/INRIA

Branch prediction feeds a speculative execution processor core with instructions. Branch mispredictions are inevitable and have negative effects on performance and energy consumption. With the advent of highly accurate conditional branch predictors, unconditional branch instructions are gaining importance.

In this article, we address the prediction of procedure returns. On modern processors, procedure returns are predicted through a return address stack (RAS). The overwhelming majority of the return mispredictions are due to RAS overflows and/or overwriting the top entries of the RAS on a mispredicted path. These sources of misprediction were addressed by previously proposed speculative return address stacks [Jourdan et al. 1996; Skadron et al. 1998]. However, the remaining misprediction rate of these RAS designs is still significant when compared to state-of-the-art conditional predictors.

We present two low-cost corruption detectors for RAS predictors. They detect RAS overflows and wrong path corruption with 100% coverage. As a consequence, when such a corruption is detected, another source can be used for predicting the return. On processors featuring a branch target buffer (BTB), this BTB can be used as a free backup predictor for predicting returns when corruption is detected.

Our experiments show that our proposal can be used to improve the behavior of all previously proposed speculative RASs. For instance, without any specific management of the speculative states on the RAS, an 8-entry BTB-backed up RAS achieves the same performance level as a state-of-the-art, but complex, 64-entry self-checkpointing RAS [Jourdan et al. 1996]. Therefore, our proposal can be used either to improve the performance of the processor or to reduce its hardware complexity.

Categories and Subject Descriptors: B.1 [**Hardware**]: Control Structures and Microprogramming; C.1.1 [**Processor Architectures**]: Single Data Stream Architectures

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Return address prediction, corruption detection, back-up predictor

Authors' addresses: H. Vandierendonck, Universiteit Gent, Vakgroep ELIS, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium; A. Sez nec, IRISA, Campus de Beaulieu, 35042 RENNES Cedex, France.

Permission to make digital or hard copies part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from the Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2008 ACM 1544-3566/2008/11-ART15 \$5.00 DOI 10.1145/1455650.1455654 <http://doi.acm.org/10.1145/1455650.1455654>

ACM Transactions on Architecture and Code Optimization, Vol. 5, No. 3, Article 15, Publication date: Nov. 2008.

The remainder of this paper is not included as this paper is copyrighted material. If you wish to obtain an electronic version of this paper, please send an email to bib@elis.UGent.be with a request for publication P108.190.pdf.
