RETROSPECTIVE:

Alternative Implementations of Two-Level Adaptive Training Branch Prediction

Tse-Yu Yeh

Intel Corporation,
Santa Clara, CA
tyyeh@mipos2.intel.com

Yale N. Patt

Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI 48109
patt@eecs.umich.edu

The Two-Level Adaptive branch predictor was conceived at Michigan during October, 1990. At the time, we and Mike Butler, another Michigan Ph.D. student in the HPS research group, were collaborating extensively with Mike Shebanow, Mitch Alsup, and Hunter Scales, all of Motorola, on a paper showing that Instruction Level Parallelism was greater than two [1].

The collaboration was initiated by Mike Shebanow, a designer of Motorola's MC88120. Shebanow was one of the original inventors of the HPS execution model, which attempted to obtain performance by wide-issue instruction supply and multiple deep pipelines with out-of-order execution to prevent blocking. He had shown as early as 1984 that more than 1/3 of the potential performance of an HPS microengine was lost due to branch prediction misses, and had proposed [2] his Autocorrelation Predictor as a way to improve on the saturating two-bit up-down counter [3], which was the most accurate predictor at that time. As part of that collaboration, Tse-Yu Yeh and Mike Butler worked with Shebanow at Motorola the previous summer, and Yale Patt visited Motorola regularly. Our studies that summer, based on the HPS paradigm, confirmed that the amount of work that would be thrown away due to a branch misprediction was prohibitively far too large. Thus, anything less than a very aggressive dynamic branch predictor was unacceptable.

The outgrowth of that summer resulted in the Two-Level Adaptive Branch Predictor. It was first published in Micro-24, in November 1991 [4], followed by the more comprehensive study in ISCA-1992 [5].

Tse-Yu Yeh presented the concept at the University of Michigan Industrial Affiliates meeting (IPoCSE) in Ann Arbor, in April, 1991, with representatives of Intel in attendance. At the time, Intel was already strongly considering a wide-issue, deeply pipelined implementation of the x86 architecture, and knew that the two-bit saturating counter mechanism would not provide sufficient prediction accuracy. Their reaction to the Two-level predictor was one of excitement. They subsequently adapted the model to their needs in what came to be the Pentium Pro microprocessor. The Two-Level predictor has continued to evolve since its beginnings in 1990, by its originators at Michigan and by other researchers at many major university and industrial research centers. Pan et. al. [6] introduced the GAs predictor, which took advantage of correlation among branches in the same equivalence class. McFarling [7] modified the use of the history register for indexing into the Pattern History Tables, reducing negative interference. He called his branch predictor gshare. Nair [8] suggested the History Register keep track of the history of the path of previous branches, rather than the history of their directions. Chang [9] augmented the set of Pattern History Tables of two-bit counters with a table of target addresses to handle indirect branches. Several authors have suggested combining compile-time information with the dynamic predictor. Chang [10] suggested classifying branches at compile time so that the dynamic predictor would only be used on non-unidirectional branches, reducing interference. Sechrest [11] investigated the role of adaptivity in the PAg Two-Level predictor. Young [12] proposed using profiling and code restructuring to allow static prediction while achieving prediction accuracies approaching that of a dynamic Two-Level predictor. Recently, Evers [13] has begun to study exactly how many of the branches in the History Register really contribute to predictions, and which simply get in the way.

In summary, in 1990, it was clear to us that if the HPS paradigm, with its wide-issue instruction supply and multiple deep pipelines, was to be successful, then a very accurate branch predictor would have to be developed, since at that time, none existed. The result of our work was the Two-Level predictor. Today, the Two-Level predictor has been implemented in multiple commercial microprocessors, and branch prediction papers extending the predictor appear at virtually every major conference from research groups at many major universities.

Tse-Yu Yeh received his Ph.D. from Michigan in EECS in 1993 and has been at Intel since then. He is currently a microarchitecture manager working in the Merced project. Yale Patt continues to teach both freshmen and graduate students and direct the research of Ph.D. students at Michigan in high performance computer implementation. Mike Shebanow is now CTO and Vice President of HAL Computer Systems in Campbell, CA, where he is responsible for the development of very aggressive high performance microprocessors. The early research was supported by Motorola, NCR and Intel. Particular acknowledgment is due to Dave Mothersole of Motorola, Lee Hoevel, formerly of NCR, and Fred Pollack, Konrad Lai and Bob Colwell of Intel for believing in and supporting the early work.

References

[1] M.Butler, T-Y Yeh, Y. Patt, M. Alsup, H. Scales, and M. Shebanow, "Single Instruction Stream Parallelism is Greater than Two," *Proc. 18th International Symposium on Computer Architecture*, May, 1981.

[2] Michael C. Shebanow, "Autocorrelation Branch Prediction," unpublished technical report, 1984.

[3] James E. Smith, "A Study of Branch Prediction Strategies," *Proc. 8th International Symposium on Computer Architecture*, pp. 135-148, 1981.

[4] Tse-Yu Yeh and Yale N. Patt, "Two-Level Adaptive Branch Prediction," *Proc. 24th International Symposium on Microarchitecture*, pp. 51-61, 1991.

[5] Tse-Yu Yeh and Yale N. Patt, "Alternative Implementations of Two-Level Adaptive Branch Prediction," *Proc. 19th International Symposium on Computer Architecture*, pp. 124-134, 1992.

[6] S.-T. Pan, K. So and J. T. Rahmeh, "Improving the Accuracy of Dynamic Branch Prediction Using Branch Correlation," *Proc. 5th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 76-84, 1992.

[7] Scott McFarling, *Combining Branch Predictors*, Technical Report, TN-36, Digital Western Research Laboratory, June, 1993.

[8] Ravi Nair, "Dynamic Path-Based Branch Correlation," *Proc. 28th International Symposium on Microarchitecture*, pp. 15-23, 1995.

[9] Po-Yung Chang, Eric Hao and Yale N. Patt, "Predicting Indirect Jumps using a Target Cache," *Proc. 24th International Symposium on Computer Architecture*, pp.274-283, 1997.

[10] Po-Yung Chang , Eric Hao, Tse-Yu Yeh and Yale N. Patt, "Branch Classification: A New Mechanism for Improving Branch Predictor Performance," *Proc. 27th International Symposium on Microarchitecture*, pp.22-31, 1994.

[11] Stuart Sechrest, Chih-Chieh Lee and Trevor Mudge, "The Role of Adaptivity in Two-Level Adaptive Branch Prediction", *Proc. 28th International Symposium on Microarchitecture*, pp. 264-269, 1995.

[12] Cliff Young and Michael D. Smith, "Improving the Accuracy of Static Branch Prediction Using Branch Correlation," *Proc. 6th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp.232-241, 1994.

[13] Marius Evers, Sanjay J. Patel, Robert S. Chappell and Yale N. Patt, "An Analysis of Correlation and Predictability: What Makes Two-Level Branch Predictors Work", *Proc. 25th International Symposium on Computer Architecture, 1998.*