Tagless Two-level Branch Prediction Schemes

I-Cheng K. Chen, Chih-Chieh Lee, Matthew A. Postiff, and Trevor N. Mudge

Technical Report CSE-TR-306-96

EECS Department, University of Michigan
1301 Beal Ave., Ann Arbor, Michigan 48109-2122
{icheng, leecc, postiffm, tnm}@eecs.umich.edu

September, 1996

Abstract

Per-address two-level branch predictors have been shown to be among the best predictors and have been implemented in current microprocessors. However, as the cycle time of modern microprocessors continue to decrease, the implementation of set-associative per-address two-level branch predictors will become more difficult. In this paper, we revisit and analyze an alternative tagless, direct-mapped approach which is simpler, requires lower power, and has faster access time. The tagless predictor can also offer comparable performance to current set-associative designs since removal of tags allows more resources to be allocated for the predictor and branch target buffer (BTB). Further, removal of tags allows decoupling of the per-address predictors from the BTB, allowing the two components to be optimized individually. We show that tagless predictors are better than tagged predictors because of opportunities for better miss-handling.

Finally, we examine the system cost-benefit for tagless per-address predictors across a wide design space using equal-cost contours. We also study the sensitivity of performance to the workloads by comparing results from the Instruction Benchmark Suite (IBS) and SPEC CINT95. Our work provides principles and quantitative parameters for optimal configurations of such predictors.

1. Introduction

As the design trends of modern microprocessors move toward wider instruction issue and deeper pipelines, effective branch prediction becomes essential to exploring the full performance of microprocessors. A good branch prediction scheme can increase the performance of a microprocessor by eliminating the instruction fetch stalls in the pipelines. As a result, numerous high performance branch prediction schemes have been proposed, such as two-level adaptive branch predictors [Yeh91], correlation-based predictors [Pan92, Yeh92b], and hybrid branch predictors [McFarling92, Chang94].

Among different predictors proposed, the two-level per-address branch predictor has been shown to be one of the best and has been implemented in the Intel Pentium Pro processor [MReport95]. Typically, the two-level per-address predictor is coupled with a branch-target buffer (BTB) through the sharing of common tags [Yeh92a, Calder94]. Both components benefit from tags and, thus, cost can be reduced by sharing. In particular, the tags enable high hit-rate set-associativity design for both history entries in predictor and BTB.

However, as the clock frequency of modern microprocessors continues to increase, the coupled set-associative design using tags may no longer be the best choice. This is because set-associative designs require longer access time than direct-mapped designs and, thus, may become a critical path in a high clock rate microprocessor. In addition, the tag-comparison in set-associative designs requires extra power. Therefore, we re-evaluate and suggest an alternative tagless direct-mapped version of two-level per-address predictors [Yeh91].

A tagless direct-mapped per-address predictor may offer performance comparable to current implementations. Although the tagless predictor does not have high hit-rate as a set-associative design does, it offers two advantages. First, by removing expensive tag storage, more resources can be allocated to the predictor and BTB, to improve performance. Second, by decoupling the BTB from the predictor, the tagless design offers the flexibility to optimize the BTB and predictor individually. In particular, the predictor can have different number of entries than the BTB. Thus, the BTB need only store taken branches instead of all branches [Calder94].

We further show that, for the prediction process, tagless predictors in general perform better, or no worse, than tagged predictors. To analyze the improvement, we break down the total errors into transitional-state and steady-state errors. Then we show that tagless predictors have lower transitional errors due to a better miss-handling policy and, consequently, have higher performance. Moreover, the tagless predictor is simpler and faster than the tagged version.

We then develop general design principles for tagless predictors. By exploring a large part of the design space, we derive general principles for selecting the best parameters. When given a specific budget and benchmark suite, these principles can help designers to select the best configurations.

The rest of this paper is organized as follows. In Section 2 we briefly review the two-level per-address predictors. In Section 3 we discuss the tagless per-address predictor scheme. In Section 4 we explain why the tagless scheme can have a better prediction accuracy than a traditional tagged scheme. Section 5 develops a cost analysis procedure to identify optimal tagless predictor designs. We present some concluding remarks with Section 6.

2. Two-level per-address adaptive branch predictors

The two-level per-address adaptive branch predictor is a variation of two-level branch predictors proposed by Yeh and Patt [Yeh91, Yeh92b]. As shown in Figure 1, a two-level per-address adaptive branch predictor consists of two tables. The first-level table, called the branch history table (BHT), has multiple shift-registers called branch history registers (BHRs). Each of these registers is used to record past branch outcomes for a single static branch. The branch outcome patterns recorded in the first-level table are then used to index a set of counters in the second level. The column index into the counters is usually some part of the address of the branch being predicted. Although there are many options for the counters, the best performance has been observed when the counters are two-bit saturating up-down counters [Nair95], and this fact was analyzed by Chen et al. [Chen96].

Since the counters are typically organized into a two-dimensional array, there can be many configurations for the second-level table. If a configuration has multiple rows and columns, then it is generally referred to as a PAs scheme according to the taxonomy by Yeh and Patt [Yeh92b]. If the table has a single column, it is a PAg scheme. If the table is a single row, the predictor is equivalent to the traditional two-bit counter scheme proposed by Smith [Smith81] because the counters are exclusively indexed by the branch address. This design space has been thoroughly studied by Sechrest et al. [Sechrest96].

The two-level per-address predictor has been shown to be among the best predictors currently in use. It has also been adopted in industry for high performance microprocessors. For

Adobe’s PostScript Language Reference Manual, 2nd Edition, section H.2.4
says your EPS file is not valid, as it calls setpagedevice