Improving Dynamic Branch Prediction Accuracy by Variable Cross-Reference Prediction and Iterative Dispatch

Po-Jen Chuang, Yue-Ter Liau, Young-Tzong Hsiao, and Yu-Shian Chiu

Department of Electrical Engineering, Tamkang University

E-mail: pjchuang@ee.tku.edu.tw

Abstract

This paper presents two schemes, the variable cross-reference (VCR) prediction scheme and the iterative dispatch approach, to improve branch prediction accuracy for the two-level adaptive branch predictor. ***The VCR prediction scheme*** makes predictions by variably cross-referring traces in the pattern history table (PHT); it achieves desirable prediction accuracy with low time complexity and at no extra hardware cost. ***The iterative dispatch approach*** enhances prediction accuracy using the PHT history to do dispatching for an additional layer of pattern history. The proposed VCR scheme is then joined by the optimal PPM algorithm to form a combined predictor which maintains desirable prediction accuracy at reduced cost. Extensive trace-driven simulation runs have been conducted to evaluate the performance of our proposed schemes and other predictors.

Keywords: Dynamic branch prediction, performance evaluation, prediction accuracy, trace-driven simulation, two-level adaptive branch predictor.

1. Introduction

Branch prediction is important in maintaining processor performance. As high prediction accuracy ensures better performance, raising prediction accuracy becomes essential. A number of new schemes, such as the 2-bit counter [1], the Markov predictor [2], the PPM algorithm [2], and the gshare [3], agree [4], bi-mode [5], YAGS [6] and DHLF predictors [7], are built to lift up prediction accuracy for the two-level adaptive branch predictor [8] in recent years. Each scheme has certain limitations. For instance, the 2-bit counter and the Markov predictor fail to provide adequate prediction accuracy for high performance processors, and the PPM algorithm yields remarkable prediction accuracy while involves considerably high complexity.

The goal of this paper is to improve branch prediction accuracy at reasonably low cost. A new prediction scheme, called the variable cross-reference (VCR) prediction scheme, is first established to deal with the prediction part of a two-level adaptive branch predictor. The proposed VCR scheme makes desirable predictions in terms of prediction accuracy, time complexity and hardware cost by variably cross-referring to traces in the PHTs. It makes use of the loop history existing in programs to elevate the prediction accuracy. An iterative dispatch approach which involves structural changes in the dispatch part of the branch prediction is also proposed in the paper. In the approach, the PHT functions as an intermediary index tag stage, the branch history in each PHT entry is used as an index tag indexing to an entry in the corresponding sub-PHT at an additional stage, and the branch history in the indexed sub-PHT entry is then used for prediction. The iterative approach thus helps divide information into more classes to reduce the PHT interference and to enhance prediction accuracy accordingly. A hybrid predictor combining the proposed VCR scheme and the optimal PPM algorithm is also introduced to attain desirable performance with reduced complexity. Extensive trace-driven simulation runs using the SPEC CINT95 benchmarks [9] have been conducted to evaluate and compare the performance of our proposed schemes and other related schemes.

2. The Variable Cross-Reference (VCR) Prediction Scheme

Different from previous schemes dealing with the prediction part, our proposed VCR scheme involves the loop history which, existing in most programs and easily observable in small programs, can execute a large quantity of branch instructions, and is helpful to elevate the accuracy of branch prediction.

In the PHT, the second level of the two-level dynamic predictor, a prediction scheme is used to predict the outcome of a branch according to the sequence of branch outcomes (taken or not taken — represented by a single bit 1 or 0) in the addressed PHT entry. The proposed VCR prediction scheme operates as follows. The sequence of outcomes in the

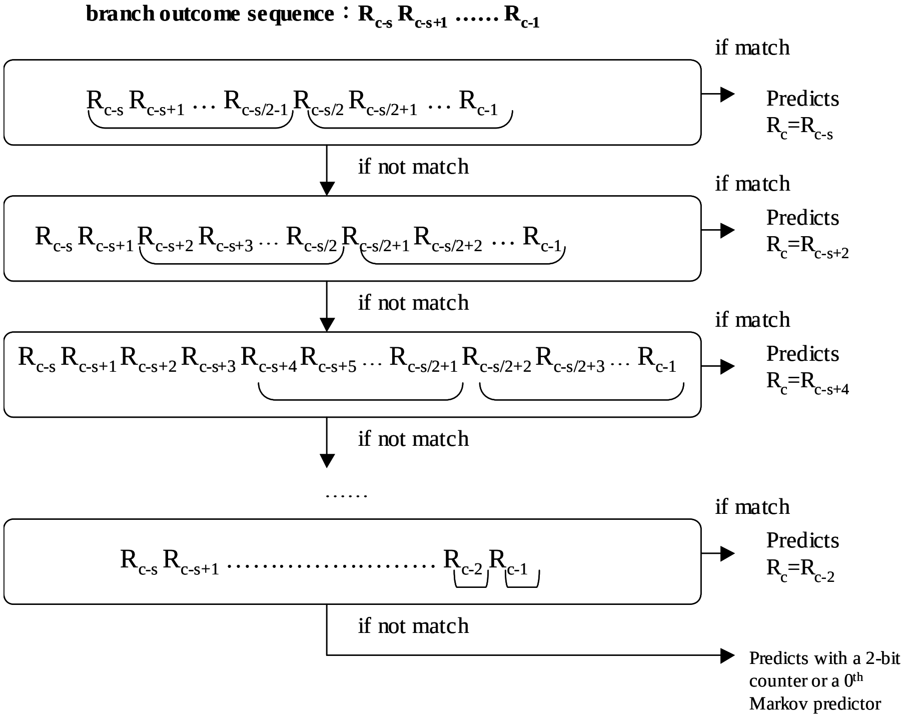

branch outcome sequence : R_{c-s} R_{c-s+1} …… R_{c-1}
R_{c-s} R_{c-s+1} … $R_{c-s/2-1}$ $R_{c-s/2}$ $R_{c-s/2+1}$ … R_{c-1}
if match
Predicts $R_c=R_{c-s}$
if not match
R_{c-s} R_{c-s+1} R_{c-s+2} R_{c-s+3} … $R_{c-s/2}$ $R_{c-s/2+1}$ $R_{c-s/2+2}$ … R_{c-1}
if match
Predicts $R_c=R_{c-s+2}$
if not match
R_{c-s} R_{c-s+1} R_{c-s+2} R_{c-s+3} R_{c-s+4} R_{c-s+5} … $R_{c-s/2+1}$ $R_{c-s/2+2}$ $R_{c-s/2+3}$ … R_{c-1}
if match
Predicts $R_c=R_{c-s+4}$
if not match
……
R_{c-s} R_{c-s+1} …………………… R_{c-2} R_{c-1}
if match
Predicts $R_c=R_{c-2}$
if not match
Predicts with a 2-bit counter or a 0th Markov predictor

Fig. 1. Prediction flowchart of our VCR scheme.

addressed PHT entry is first divided into two parts that are equal in length, i.e., number of outcomes. (If there is an odd number of outcomes in the PHT entry, the "least recent" outcome could be ignored.) The two parts are then cross-referred to see if they match with each other. If both parts are completely the same, we assume the same outcome will repeat again (according to the loop history) and thus predict the coming branch outcome to be the first outcome in the first part (also the first outcome in the second part). If the two parts do not match, ignore the next two "least recent" outcomes in the sequence and again divide the remaining outcomes into two equal-length parts. Check the two parts: If they are the same, predict the outcome to be the first outcome of the first part; if not, repeat the above referring process until a match for the two parts is located. In case no match is found when the number of referred outcomes is reduced to only one in each part, employ some other scheme (such as the 2-bit counter or 0th Markov predictor) to assist the prediction.

Fig. 1 demonstrates the operation of the proposed scheme. The sequence of branch outcomes in the addressed PHT, assumed to be $R_{c-s}R_{c-s+1} \dots R_{c-1}$ with s (the number of outcomes in each PHT entry) being an even number, is first divided into two equal-length parts, i.e., $R_{c-s}R_{c-s+1} \dots R_{c-s/2-1}$ and $R_{c-s/2}R_{c-s/2+1} \dots R_{c-1}$. The two parts are then compared. If they match each other, that is, if

$$R_{c-s} = R_{c-s/2},\ R_{c-s+1} = R_{c-s/2+1},\ \dots \text{ and } R_{c-s/2-1} = R_{c-1},$$

the coming branch outcome is predicted to be $R_c = R_{c-s}$. If they do not match each other, ignore the two "least recent" outcomes R_{c-s} and R_{c-s+1} and divide the remaining outcomes into two new parts $R_{c-s+2}R_{c-s+3} \dots R_{c-s/2}$ and $R_{c-s/2+1}R_{c-s/2+2} \dots R_{c-1}$. Check again. If the two parts match, i.e., if

$$R_{c-s+2} = R_{c-s/2+1},\ R_{c-s+3} = R_{c-s/2+2},\ \dots \text{ and } R_{c-s/2} = R_{c-1},$$

our prediction will be $R_c = R_{c-s+2}$. If they do not match, ignore the next two "least recent" outcomes R_{c-s+2} and R_{c-s+3}, and again divide the remaining outcomes into $R_{c-s+4}R_{c-s+5} \dots R_{c-s/2+1}$ and $R_{c-s/2+2}R_{c-s/2+3} \dots R_{c-1}$. If the two parts match each other, the coming branch outcome is predicted to be $R_c = R_{c-s+4}$. If they do not match, repeat the same comparison process (by ignoring the next two "least recent" outcomes at each comparison attempt). Prediction can be made whenever a match is found by this variable cross-reference. (Note that in our scheme the two parts under comparison are with variable, not fixed, lengths.) If eventually only one outcome is left in each part and they still do not match each other, the prediction is handed over to a 2-bit counter or a 0th Markov predictor. It is based on such a variable cross-reference process (which needs no extra hardware at all) that the proposed

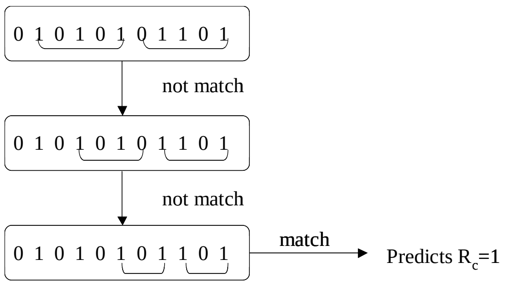


Fig. 2. Example of our VCR scheme

prediction scheme is called the Variable Cross-Reference (VCR) scheme.

Fig. 2 further illustrates the VCR scheme. As it shows, there are 11 bits (01010101101) in the addressed PHT entry. Ignore the most significant bit, i.e., the "least recent" branch outcome, and divide the remaining 10 bits into two equal-length parts 10101 and 01101. Compare the two parts. As there is no match, ignore the next two "least recent" bits and cross-refer the newly divided two parts 1010 and 1101. Since there is no match, ignore the next two "least recent" bits and divide the remaining bits (101101) into two new parts 101 and 101. With the two parts matching each other, we thus predict the coming branch outcome to be '1' (i.e., the most significant bit in both parts).

3. The Iterative Dispatch Approach

The iterative dispatch approach involves some structural changes in the dispatch part of a two-level adaptive branch predictor. In our new design, the PHT functions as an intermediary index tag stage and the branch history in each PHT entry is used as an index tag indexing to an entry in the corresponding sub-PHT at an additional stage. The branch history in the indexed sub-PHT entry is then used for prediction. For instance, if the bits in the branch history register (BHR) are $R_{c-k}R_{c-k+1} \ldots R_{c-1}$, it will address an entry in the PHT. However, the history bits in the addressed PHT entry are used not for prediction but as an index tag indexing the corresponding sub-PHT at the additional stage. Suppose the length of the PHT entry is m bits, we can index to a corresponding sub-PHT with 2^m entries in the same way as indexing the BHR to the PHT, and have 2^{m+k} sub-PHT entries in total. Predictions are then made by referring to the bits in the sub-PHT entries. The BHR, PHT and sub-PHTs will update their contents — after the outcome of each branch turns out — to lead the sub-PHTs for future predictions. In this way the "traces of traces" are referred to as a kind of information to improve prediction accuracy. That is, this iterative dispatch approach utilizes the PHT history to do dispatching for an additional layer of pattern history and the information can hence be further divided into 2^m classes, providing more information and less PHT interference than employing only the traditional PHTs in making predictions.

Fig. 3 exhibits the structure of our proposed iterative dispatch approach. As shown here, the PHT exists between the BHR and the sub-PHTs as an intermediary index tag stage. Data in the BHR are first classified by the intermediary index tag stage (the PHT) which is much shorter than the entire sub-PHTs. Based on the behavior of the branch outcomes in a sub-PHT entry, predictions are then made. (Note that due to such a structural change, the number of table entries increases and so does the needed warm-up time.) Assume the length of the BHR is k bits. It can address the PHT with 2^k tags and each tag indexes to an entry of the corresponding sub-PHT with 2^m entries. Before a prediction is made, an entry (i.e., an index tag) in the PHT is addressed according to the bits $R_{c-k}R_{c-k+1} \cdots R_{c-1}$ in the BHR. The index tag then addresses an entry in the corresponding sub-PHT (say sub-PHT$_x$, $0 \leqq x \leqq 2^k-1$). A prediction is finally made by referring to the bits in the indexed sub-PHT entry. If the branch result is R_c, it is then shifted into the BHR and the bits in the BHR are updated as $R_{c-k+1}R_{c-k+2} \ldots R_{c-1}R_c$. The PHT entry and the indexed sub-PHT entry are also updated by the bit R_c.

The iterative dispatch approach has been designed to assist predictors in elevating prediction accuracy. Take the proposed VCR predictor as an example. When encountered with the sequence of branch outcome 10110101, the PPM algorithm, Markov predictor and 2-bit counter will predict the next bit to be 1, while the VCR scheme will predict it to be 0. In fact, the sequence displays a loop history of 1011 with an extra 0 in the middle — a situation which may lead the VCR scheme to wrong predictions. For situations like this, the iterative dispatch approach can be brought in to help as demonstrated in Fig. 4. Assuming m = 1, we first initialize the intermediary index tag stage to be 0 and the sequence of branch outcome to be 10110101. After the BHR encounters the first two bits 10 and makes the prediction, shift the branch outcome (i.e., 1) into both entry 10 of the PHT and entry 0 of the corresponding sub-PHT (i.e., sub-PHT$_{10}$). Now the newly updated information of both the addressed PHT entry and the indexed sub-PHT entry becomes 1. Then based on the branch outcome for the next 2 bits 01, the content of the PHT entry 01 and the indexed sub-PHT$_{01}$ entry are also updated with the outcome 1. As the original content of the

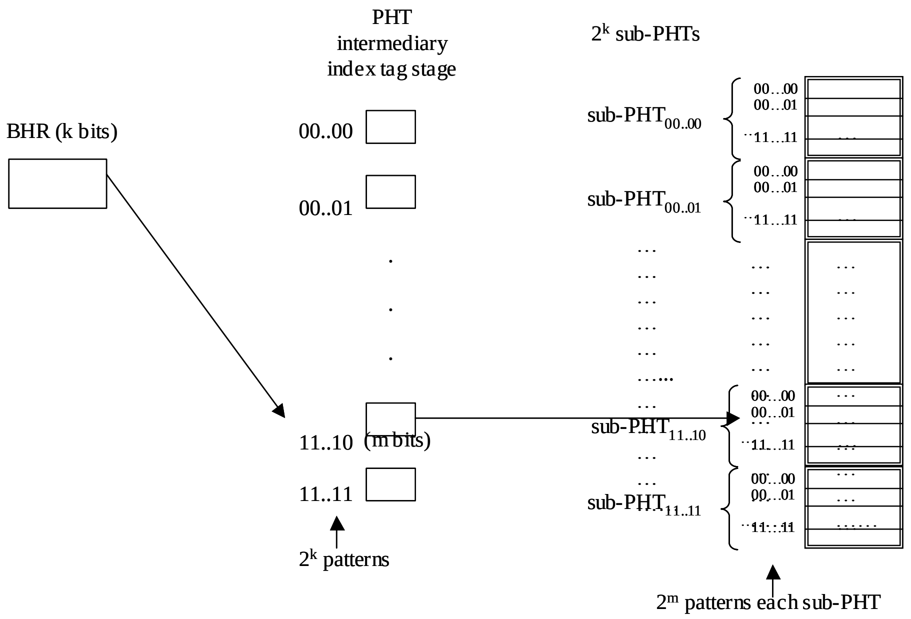


Fig. 3. Structure of the iterative dispatch.

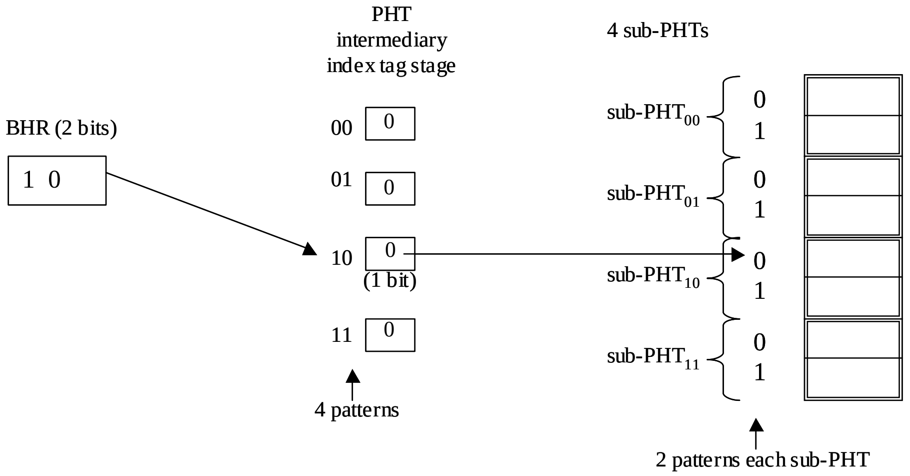


Fig. 4. Example of the iterative dispatch.

PHT entry 01 is 0, we update entry 0, instead of entry 1, of sub-PHT$_{01}$. The content is now updated to 1. Such a shifting and updating process is repeated following every two bits of the sequence until 01 is again shifted into the BHR. With the content of the PHT entry 01 being updated to 1, entry 1 of sub-PHT$_{01}$ will be updated accordingly. When the end bits of sequence 01 is shifted into the BHR, the referred PHT will be entry 0 of sub-PHT$_{01}$ because the content of the updated PHT entry 01 is 0 (due to the branch outcome after the last 2-bit sequence 01 being 0). As the content of entry 0 of sub-PHT$_{01}$ is 1, the VCR scheme will thus predict the branch outcome to be 1, like the other schemes. The proposed iterative dispatch approach is shown through simulation results to work not only for the VCR scheme but also for other schemes, especially for schemes with lesser performance, such as the 2-bit counter (to be discussed in later sections).

4. The PPM-VCR Predictor

A combined predictor is composed of at least two single predictors which simultaneously make predictions when a branch occurs. A selector is employed to evaluate the prediction performance of each (single) predictor. Based on previous outcomes, the selector will check and choose the predictor most likely to make the correct prediction for the current branch. Branch prediction outcomes made by each predictor are recorded in a 2-bit counter that updates itself with each new result. Following the continually updated data, the 2-bit counter is able to decide a better predictor and select it for predicting the incoming branch.

While the PPM algorithm attains remarkable prediction accuracy at the cost of substantial complexity, the proposed VCR scheme depicts quite satisfying prediction accuracy with much less complexity. Indeed the VCR scheme performs even better than the optimal PPM algorithm under certain conditions, such as during the warm-up period or with shorter PHTs. (When the PHTs are short, "referring" tends to yield the same probability for "taken" and "not taken" of the branch, making the PPM algorithm unable to make correct predictions. The VCR scheme is free of such limitations. It can make fast and correct predictions whenever the cross-reference finds a match.) We are thus interested in combining the two prediction schemes together to see the performance of the combined predictor. For the combined PPM-VCR predictor, we choose not to use a 2-bit counter as the priority selector considering the performance and overhead of the two prediction schemes. Instead, the priority selector for the PPM algorithm is expanded into an (n-1)-bit counter (the length of the PHT is 2^n) and that for the VCR scheme is set to be a 1-bit counter. When making predictions, employ the PPM algorithm to do the job if it displays larger priority; otherwise, employ the VCR scheme. The priority selectors are updated with each new prediction result for future predictions.

5. Performance Evaluation

Extensive trace-driven simulation runs using four SPEC CINT95 benchmarks [9] — vortex, perl, m88ksim and gcc — are conducted to evaluate the performance of our proposed schemes and other schemes. The SimpleScalar Toolset [10] is used to generate and capature address traces. Prediction accuracy is the performance measure of interest, but for more informative presentation, misprediction rates (one minus prediction accuracy) are presented in our discussions, as in [5]. Note that the Markov predictor is not included in the simulation because of its prediction limitations for zero and equal frequencies, and the PPM algorithm adopted here is the optimal one. The misprediction rates are collected under various BHR lengths (2 ~ 7 bits) and PHT lengths (8 ~ 256 bits). Due to very limited space, only the misprediction rates collected under PHT lengths = 8 ~ 256 bits with BHR length = 7 bits, and under BHR lengths = 2 ~ 7 bits with PHT length = 256 bits are presented for performance comparisons (1) and (2).

(1) Predictors dealing with the prediction part

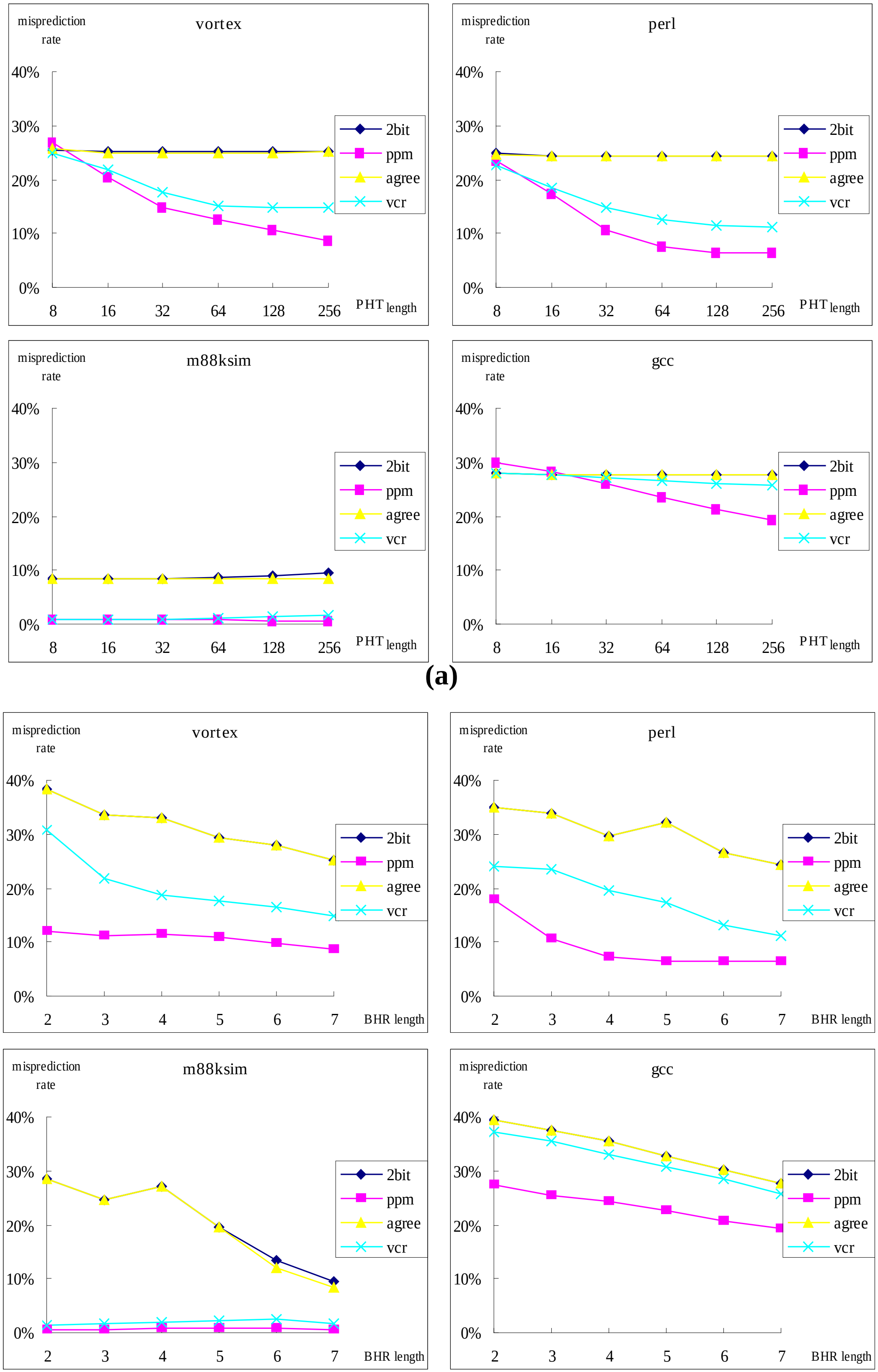
Depicted in Fig. 5(a) are the misprediction rates for the 2-bit counter, the PPM algorithm, the agree predictor and the VCR scheme resulting from running the four SPEC CINT benchmarks under PHT lengths = 8 ~ 256 bits with BHR length = 7 bits (a similar performance trend can be found with any of the BHR lengths). As exhibited, the performance of our VCR scheme yields constantly lower misprediction rates than the 2-bit counter and the agree predictor. In fact, the proposed scheme outperforms even the optimal PPM algorithm at shorter PHTs, such as 8 bits, in some benchmarks. This is because with shorter PHTs, "referring" for the PPM algorithm tends to yield the same probability for "taken" and "not taken" of the branch, making the algorithm unable to predict correctly. By contrast, misprediction rates for the PPM algorithm at longer PHT lengths are apparently lower than that for the 2-bit counter, the agree predictor and the VCR scheme. It should nevertheless be pinpointed that the high performance of the PPM algorithm is achieved at substantial cost as our simulation adopts the largest predictable PHT length — 256 bits, which enables the PPM algorithm to use the 255th PPM predictor or 256 Markov predictors to predict the branches. The misprediction rates of these schemes collected under BHR lengths = 2 ~ 7 bits with PHT length = 256 bits in Fig. 5(b) exhibit a similar trend as what is shown in Fig. 5(a).

(2) Predictors dealing with the dispatch part

Performance of the gshare predictor, the DHLF predictor and our iterative dispatch approach is illustrated in Fig. 6(a) where the misprediction rates are collected under various PHT lengths with BHR length = 7 bits. The figures show that misprediction rates obtained from the four benchmarks are always lower for our iterative dispatch approach than for the other two schemes. This is because the iterative dispatch approach utilizes the PHT history to do dispatching for an additional layer of pattern history and by dividing the information into more classes, it is able to provide more information and reduce the PHT interference when making predictions. Similar results can be found in Fig. 6(b) which depicts misprediction rates under various BHR lengths with PHT length = 256 bits.

(3) The VCR scheme, the bi-mode predictor and the YAGS predictor

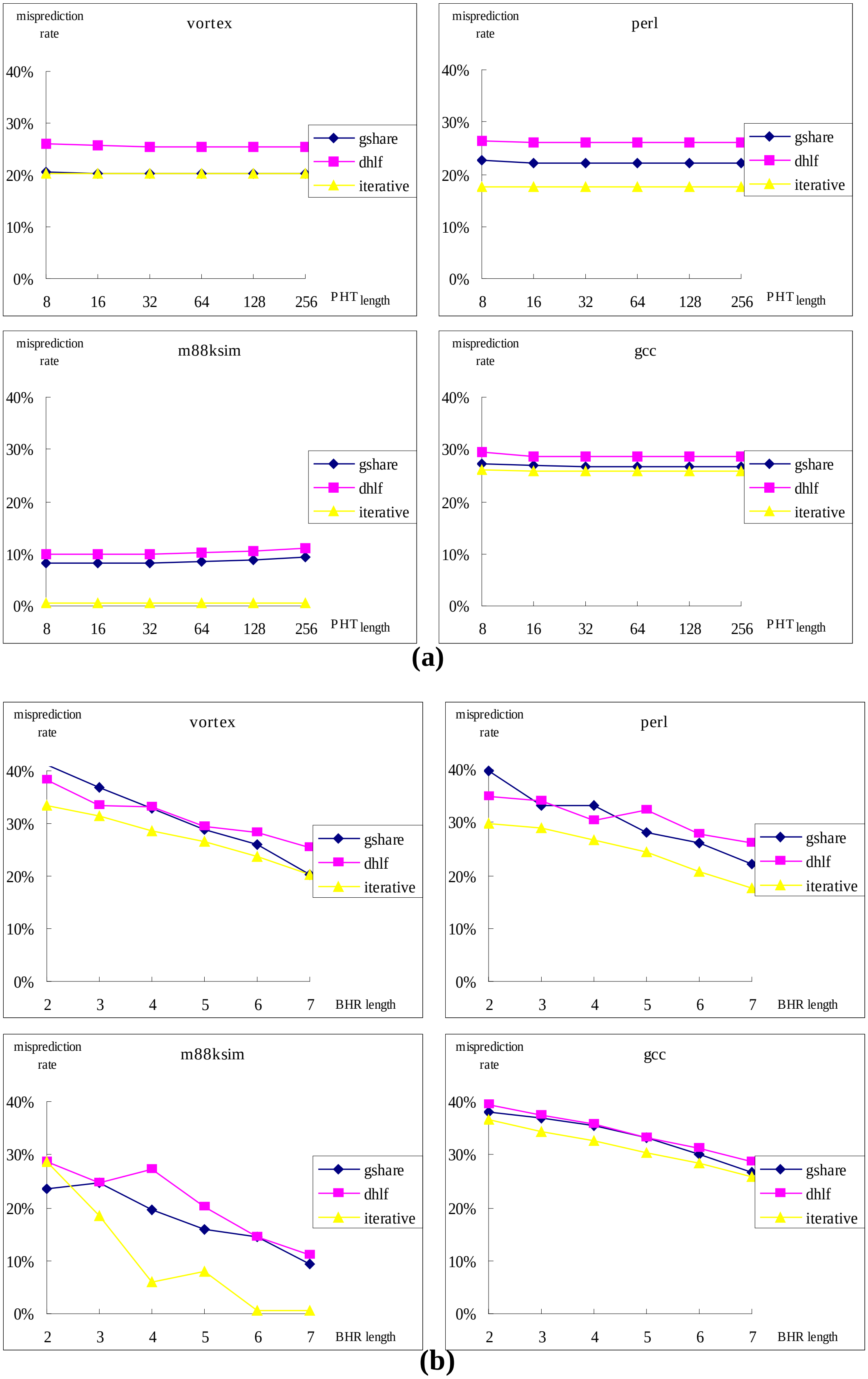
Simulation results show that the overall performance of our VCR scheme excels that of the other 2 schemes in all benchmarks except



(a)

(b)

Fig. 5. Misprediction rates for schemes dealing with the prediction part.



(a)

(b)

Fig. 6. Misprediction rates for schemes dealing with the dispatch part.

benchmark gcc where the VCR scheme falls behind the bi-mode predictor and also the YAGS predictor in some situations — with slight differences. In more practical situations, such as with longer PHT or BHR lengths, the performance of our VCR scheme compares favorably, at no extra cost, to the other two predictors which need more extra hardware and cost (such as two extra direction PHTs and doubled predictions in the choice and direction PHTs for the bi-mode predictor).

(4) The PPM-VCR predictor

The performance of the VCR scheme, the PPM algorithm and the combined PPM-VCR predictor is also simulated. As the results indicate, the PPM-VCR predictor performs as well as or better than the PPM algorithm alone. This is especially significant when we take the potentially reducible complexity of the VCR scheme into account.

6. Conclusion

To improve branch prediction accuracy, a variable cross-reference (VCR) prediction scheme and an iterative dispatch approach are proposed in this paper. The proposed VCR scheme can be easily implemented and is able to yield desirable prediction accuracy for a high performance processor at low cost. To further enhance prediction accuracy, an iterative dispatch approach is provided. The approach utilizes the PHT history to do dispatching for an additional layer of pattern history which helps providing more information for making better predictions. It is shown that the proposed VCR scheme and iterative dispatch approach can handily work with other predictors to fortify performance. A PPM-VCR combined predictor is also presented to demonstrate the advantages of a combined predictor.

Simulation results show that the overall performance of our VCR scheme compares favorably to other schemes, such as the 2-bit counter and the agree predictor due to its variable cross-reference to the traces in the PHT. With much less complexity, the VCR scheme even outperforms the optimal and yet complicated PPM algorithm under some conditions. When compared with the bi-mode and YAGS predictors — which deal with both the prediction and dispatch parts of the two-level predictor and require extra hardware and cost, the VCR scheme still produces better performance in most of the situations. The proposed iterative dispatch approach is shown through experimental evaluation to outperform the gshare and DHLF predictors, schemes dealing with the dispatch part. It also lifts prediction accuracy for different schemes, especially for schemes with lesser performance, such as the 2-bit counter. In contrast to the performance of the optimal PPM algorithm, slight degrees of improvement can be detected for the performance of the PPM-VCR combined predictor. The performance gain alone may not appear significant enough, but the potentially reducible complexity due to the VCR scheme is nevertheless appealing.

References

[1] T.-Y. Yeh and Y. N. Patt, “Alternative implementations of two-level adaptive branch prediction,” *Proc. 19th Annual Int'l Symp. on Computer Architecture*, May 1992, pp. 124-134.

[2] I-C. K. Chen, J. T.Coffey, and T. N. Mudge, “Analysis of branch prediction via data compression,” *Proc. 7th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, Oct. 1996, pp. 128-137.

[3] S. McFarling, “Combining branch predictors,” *Technical Report, TN-36*, Digital Western Research Laboratory, June 1993.

[4] E. Sprangle, R. S. Chappell, M. Alsup, and Y. N. Patt, “The agree predictor: A mechanism for reducing negative branch history interference,” *Proc. 24th Annual Int’l Symp. on Computer Architecture*, May 1997, pp. 284-291.

[5] C.-C. Lee, I-C. K. Chen, and T. N. Mudge, “The bi-mode branch predictor,” *Proc. 30th Int’l Symp. on Microarchitecture*, Dec. 1997, pp. 4-13.

[6] A. N. Eden and T. Mudge, “The YAGS branch prediction scheme,” *Proc. 31st Int’l Symp. on Microarchitecture*, Dec. 1998, pp. 69-77.

[7] T. Juan, S. Sanjeevan, and J. J. Navarro, “Dynamic history-length fitting: A third level of adaptivity for branch prediction,” *Proc. 25th Annual Int’l Symp. on Computer Architecture*, May 1998, pp. 155-166.

[8] T.-Y. Yeh and Y. N. Patt, “Two-level adaptive branch prediction,” *Proc. 24th Annual Int'l Symp. on Microarchitecture*, Nov. 1991, pp. 51-61.

[9] *SPEC CPU’95, Technical Manual*, Aug. 1995.

[10] D. Burger and T. M. Austin, “The SimpleScalar Tool Set, Version 2.0,” *Technical Report #1342*, Univ. of Wisconsin-Madison CS Dept. June 1997.