CS204: Lab 6 - Branch Prediction Report

Objective: The objective of this lab is to gain an understanding of branch prediction accuracy using different methods.

Introduction: Branch prediction plays a crucial role in modern computer architecture, particularly in optimizing the performance of processors. By predicting the direction of conditional branches, processors can pre-fetch instructions and minimize pipeline stalls. In this lab, we explore four branch prediction algorithms: Always Taken, Always Not Taken, 1-bit Dynamic Branch Predictor, and 2-bit Dynamic Branch Predictor.

Methods: We extended the assembler developed in the mini-project of this course to read the dynamic instruction trace. The format of instructions was modified to replace labels in branch instructions with integers. For each instruction, we predicted the next instruction address if the instruction is a branch instruction. We implemented the following branch prediction methods:

1. **Always Taken:** Predicts that the branch will always be taken.
2. **Always Not Taken:** Predicts that the branch will never be taken.
3. **1-bit Dynamic Branch Predictor:** Maintains a single bit per branch instruction, indicating whether it was taken or not last time.
4. **2-bit Dynamic Branch Predictor:** Maintains a two-bit saturating counter per branch instruction, which predicts whether the branch will be taken based on past behavior.

We maintained a branch target buffer and a history table in the assembler to facilitate branch prediction.

Testing: We utilized the following programs for testing our branch prediction algorithms:

1. Bubble Sort
2. Factorial
3. Binary Search

Results:

1. **Bubble Sort:**
    - Always Taken Accuracy: 41.4823%
    - Always Not Taken Accuracy: 58.5177%
    - 1-bit Dynamic Branch Predictor Accuracy: 97.1078%
    - 2-bit Dynamic Branch Predictor Accuracy: 97.1078%

- Branch Target Buffer: In output File “Bubble_test_Lab_OutputBTB.txt”.
- History Table: In Output File “Bubble_test_Lab_OutputHT.txt”.

2. Factorial:

- Always Taken Accuracy: 63.7548%
- Always Not Taken Accuracy: 36.2452%
- 1-bit Dynamic Branch Predictor Accuracy: 88.39%
- 2-bit Dynamic Branch Predictor Accuracy: 90.3403%
- Branch Target Buffer: In output File “Fac_test_Lab_OutputBTB.txt”.
- History Table: In Output File “Fac_test_Lab_OutputHT.txt”.

3. **Binary Search:**

- Always Taken Accuracy: 47.2727%
- Always Not Taken Accuracy: 52.7273%
- 1-bit Dynamic Branch Predictor Accuracy: 67.2727%
- 2-bit Dynamic Branch Predictor Accuracy: 76.3636%
- Branch Target Buffer: In output File “Binary_search_OutputBTB.txt”.
- History Table: In Output File “Binary_search_OutputHT.txt”.

Comparison of Branch Prediction Algorithms:

1. **Bubble Sort:**

- Among the tested branch prediction algorithms, the 2-bit Dynamic Branch Predictor achieved the highest accuracy of 97.1078% for Bubble Sort. This indicates that the branch behavior within the Bubble Sort algorithm is well-suited for dynamic prediction mechanisms.
- While both static predictors (Always Taken and Always Not Taken) demonstrated lower accuracies compared to dynamic predictors, the 1-bit Dynamic Branch Predictor also showed a significant improvement over the static approaches, achieving an accuracy of 97.1078%.
- The provision of Branch Target Buffer (BTB) and History Table output files suggests that the algorithm might benefit from further analysis and optimization based on branch prediction outcomes.

2. **Factorial:**

    - For the Factorial algorithm, the 2-bit Dynamic Branch Predictor again exhibited the highest accuracy, reaching 90.3403%. This suggests that the Factorial algorithm's branch behavior also benefits from the adaptability provided by dynamic prediction mechanisms.
    - Both static predictors showed notable differences in accuracy compared to Bubble Sort, with the Always Taken predictor achieving 63.7548% accuracy and the Always Not Taken predictor achieving 36.2452% accuracy. This indicates the variability of branch behavior across different algorithms.
    - The provided BTB and History Table output files indicate potential areas for further investigation into branch prediction behavior within the Factorial algorithm.

3. **Binary Search:**

    - In the Binary Search algorithm, the 2-bit Dynamic Branch Predictor once again demonstrated the highest accuracy, achieving 76.3636%. This suggests that the Binary Search algorithm's branch behavior benefits from the adaptive nature of dynamic predictors.
    - Both static predictors exhibited accuracies closer to each other compared to the previous algorithms, with Always Taken at 47.2727% and Always Not Taken at 52.7273%. This reflects the inherently more predictable nature of the Binary Search algorithm's branch behavior.
    - The provision of BTB and History Table output files indicates potential avenues for analyzing and optimizing branch prediction performance within the Binary Search algorithm.

Overall Comparison:

- Across the three test cases, the 2-bit Dynamic Branch Predictor consistently demonstrated the highest average accuracy, with an accuracy ranging from 76.3636% to 97.1078%.
- The Bubble Sort and Factorial algorithms showed significant improvement in accuracy when using dynamic predictors compared to static predictors.
- Binary Search, being inherently more predictable, exhibited lower sensitivity to the type of branch predictor used, but still showed improvement with dynamic predictors.
- The results highlight the importance of adaptive branch prediction mechanisms, especially in scenarios with complex branch behaviors or varying execution patterns. Dynamic predictors offer better adaptability and accuracy compared to static predictors, contributing to overall performance enhancement in branch prediction.

Discussion: Based on our results, we observe that the accuracy of branch prediction varies significantly depending on the nature of the program and the prediction algorithm employed. Always Taken and Always Not Taken strategies perform adequately in some cases but poorly in others. The dynamic branch predictors show better accuracy overall, with the 2-bit predictor exhibiting the highest accuracy among the dynamic methods. However, it is essential to note that the performance may vary depending on the specific characteristics of the programs being executed.

Conclusion: In conclusion, branch prediction is a critical aspect of computer architecture that significantly impacts processor performance. Through this lab, we gained insights into the accuracy of different branch prediction algorithms across various program types. Dynamic predictors, especially the 2-bit predictor, generally outperform static strategies like Always Taken or Always Not Taken. However, further optimization and fine-tuning may be necessary to achieve even higher accuracy in real-world scenarios.

References:

- Computer Organization and Embedded Systems by Carl Hamacher

Acknowledgments:

- Venus (https://venus.cs61c.org/)

Group Members:

- Dhruv Gupta (2022CSB1079)
- Kush Mahajan (2022CSB1089)
- Nishant Patil (2022CSB1097)

Submission Details:

- Complete code
- Testcases used
- Report comparing four branch prediction algorithms for the test cases
- GitHub Repository Link: Branch Predictor GitHub Repository

Date of Submission: 26/03/2024