

CSE 240A Project Report: Branch Prediction

Ke Wan, Xi Cai
Department of Computer Science & Engineering
University of California San Diego

Abstract—This document is intended to serve as a sample for submissions to the 47th IEEE/ACM International Symposium on Computer Architecture (ISCA), May 30 – June 3, 2020 in Valencia, Spain. This document provides guidelines that authors should follow when submitting papers to the conference. This format is derived from the IEEE conference template IEEEtran.cls file with the objective of keeping the submission similar to the final version, i.e., the IEEEtran.cls template will also be used for the camera-ready version.

I. INTRODUCTION

Branch predictor plays an important role on instruction sets. About 20% of all computer instructions are branch instructions. However, branch instructions are different from other kinds of instructions. There are multiple outcomes for a branch instruction. For example, for an **if-else** branch, there are two outcomes for this branch. If the value of the condition statement is true, then the program will execute the statements in the **if** code block. If not, the program will execute the statements in the **else** code block. Predicting the outcome of branches correctly is important to accelerate the speed of instruction execution and reduce the overhead time. Therefore, branch prediction strategy plays an important role in this task.

In this paper, we introduce three branch predictors, G-Share branch predictor, Tournament branch predictor, and custom Perceptron branch predictor. We discuss the structure, the design ideas, the implementation details of these three branch predictor.

G-Share predictor is the most simple branch predictor in this paper. It fetched the lowest 13-bit Program Counter(PC) and Global History Address(GHA). Then we did an XOR calculation for these two variables and map the result to Prediction History Table(PHT) to fetch the 2-bit prediction result.

Tournament branch predictor is a hybrid predictor. It contains a local branch predictor, a global branch predictor, and a choice predictor. Firstly, the Tournament branch predictor will fetch the lowest 10 bits of global history as the index of choice prediction. Then the branch predictor will determine which branch predictor should be used. If the branch predictor chooses to use the local predictor, then it will firstly fetch the lowest 9-bit PC as the index of Local History Table(LHT), then it will fetch the 10-bit Local History(LH) from LHT as the index of 2-bit local prediction result. If the branch predictor chooses the global predictor, then it will firstly get the lowest 10 bits global address as the index of global prediction table. Then it will fetch the 2-bit prediction result from global PHT.

Our custom Perceptron branch predictor is based on Perceptron Predictor [1], which contains 16 Preceptions to determine the prediction result. It contains a Table of Preception to store the perceptron weights for different local history records and a register to store global history record. When the predictor encounters a branch, it will get the lowest 9 bits of PC address as the index of Perceptron Table. Then it will fetch the weights from the table and do the prediction calculation using the formula $y_{out} = w_0 + \sum_{i=1}^n x_i w_i$. If the value of y_{out} is equal or greater than zero, then the branch predictor will predict it as TAKEN. Otherwise, the branch predictor will predict it as NOT-TAKEN.

This paper also did experiments to test the prediction accuracy of three branch predictors on six address traces provided by instructors and made comparisons quantitatively. G-Share branch predictor achieves average accuracy of **94.41%**. Tournament branch predictor achieves the average accuracy of **94.92%**, a little better than G-Share predictor. The custom Perceptron branch predictor preforms the best among these three branch predictor. It has the average accuracy of **95.29%**, and it beats the other two branch predictor in all six traces.

II. RESPONSIBILITY OF EACH MEMBER

This project is finished by two members, Ke Wan and Xi Cai. Ke Wan mainly contributes to design, develop, and test Tournament and custom Perceptron branch predictor. Xi Cai mainly contributes to develop G-share branch predictor and verified the correctness of our implementations. We cooperated to write project report.

III. BACKGROUND AND MOTIVATION

IV. DESIGN IDEAS

V. IMPLEMENTATION

VI. EXPERIMENT SETUP AND RESULTS

VII. CONCLUSION

REFERENCES

- [1] C. L. Daniel A. Jimenez, "Dynamic branch prediction with perceptrons," 2001.