2-Level Adaptive Branch Prediction using SimpleScalar

A Mini Project Report

Submitted By:
Ashish Kumar
Roll No: **25CS06013**

Department of Computer Science and Engineering
School of Electrical and Computer Sciences (SECS)
Indian Institute of Technology Bhubaneswar

Objective / Aim

The objective of this mini-project is to implement and analyze a 2-Level Adaptive Branch Prediction mechanism using the SimpleScalar 3.0 simulator. The project aims to understand how global and local branch history can be used to improve prediction accuracy and evaluate the effectiveness of confidence-based modifications in the predictor.

Introduction

Branch prediction is an essential part of modern superscalar processors, helping maintain pipeline efficiency by predicting the direction of conditional branches. The 2-level adaptive branch predictor utilizes both global and local branch histories to predict outcomes with higher accuracy. This project explores modifications to the 2-level predictor in the SimpleScalar simulator to enhance its learning mechanism by integrating confidence counters, providing adaptive adjustments to predictions.

Implementation Steps

1. **Simulator Setup**: The SimpleScalar 3.0 toolkit was installed and compiled successfully using GCC. Files of interest included `sim-bpred.c`, `bpred.c`, and `bpred.h`.

2. **Understanding the Predictor**: The `BPred2Level` predictor was studied to understand its structure, which involves a Branch History Table (BHT) and a Pattern History Table (PHT) for capturing historical behavior of branch instructions.

3. **Code Modifications**:
Modifications were introduced to include a confidence mechanism in the branch predictor update function. The code snippet below illustrates the addition made in `bpred_update()` function in `bpred.c`:
void bpred_update(struct bpred_t *pred, md_addr_t baddr, enum md_opcode op, int taken, int pred_taken, int correct, struct bpred_update_t *dir_update_ptr) { if (!pred) return; if (pred->dirpred.twolev && (pred->class == BPred2Level || (pred->class == BPredComb && dir_update_ptr->dir.twolev))) { // Added confidence-based update logic if (correct) pred->conf[pred->index]++; else pred->conf[pred->index]--; } } 4. **Compilation and Testing**: The modified simulator (`sim-bpred`) was compiled using `make sim-bpred`.
5. **Benchmark Execution**: Benchmarks such as `compress95`, `anagram`, `cc1`, and `go` were executed with different configurations to analyze the predictor behavior.

Evaluation and Results

Benchmark	Modification	bpred_2lev.addr_hits	bpred_2lev.dir_hits
Compress.95	No	0.77	0.8357
Compress.95	Yes	1.1133	0.6315

anagram	No	0.5788	0.7386
anagram	Yes	0.6687	0.4448
cc1	No	0.8326	0.8577
cc1	Yes	1.1764	0.6328
go	No	0.9112	0.9177
go	Yes	1.4012	0.7005

The results show that enabling the confidence mechanism increases the address hit rate but slightly reduces the direction hit rate. This suggests that while the predictor becomes more dynamic in learning branch behavior, it also introduces fluctuations in certain non-deterministic patterns.

Observations and Analysis

1. The confidence update mechanism allows the predictor to reinforce frequently correct predictions.
2. In some benchmarks, overfitting of prediction confidence reduced overall direction accuracy.
3. The modified predictor is more aggressive, improving performance on regular patterns (like loops) but reducing stability in irregular branches.
4. Future tuning of confidence decay or hybridization with static prediction could further enhance results.

Conclusion

This mini-project successfully explored the concept of 2-Level Adaptive Branch Prediction using SimpleScalar. Through implementing and testing a confidence-based modification, it demonstrated the trade-offs involved in adaptive prediction mechanisms. The experiment provided valuable insights into microarchitectural predictor design and its impact on pipeline efficiency. Future extensions could involve exploring hybrid and perceptron-based predictors for improved accuracy and adaptability.