Computer Assignment 2 Report

Introduction

I chose to implement the TAGE (TAgged GEometric history length) branch predictor, which is an advanced branch prediction scheme that uses multiple predictor tables indexed by differently hashed global history of branch instructions. This scheme aims to address the limitations of simpler predictors like 2-bit and gshare predictors by utilizing longer historical information and optimizing resource allocation.

Design Overview

The TAGE predictor typically consists of several tagged tables (components), each with a unique history length that follows a geometric progression. I decided not to use a geometric progression as I found I was able to achieve less mispredictions with values that I tried and tested myself. Each entry in these tables contains a prediction counter, a tag, and a useful counter. The predictor searches these tables sequentially to find matching tags and uses the prediction from the longest history table that matches. If no matches are found, a base predictor (e.g., a simple 2-bit predictor) is used.

Implementation Details

My implementation of the TAGE predictor is enclosed in the my_predictor.h file. The primary components and functionalities are as follows:

1. History Lengths: Four history lengths are initialized as 6, 12, 19, and 30.
2. Tagged Tables: Each tagged table consists of tagged entries containing a 3-bit prediction counter, 2-bit useful counter, and tag bits.
3. Base Predictor: A simple 2-bit predictor is used as a fallback when no matching tags are found.

Parameter Tuning

To achieve the best possible prediction accuracy, I experimented with various values for the parameters, specifically the history lengths, the sizes of the base and tagged tables, and the number of tag bits. After extensive testing, the following parameters were chosen:

HISTORY_LENGTH: 30

BASE_TABLE_BITS: 19

TAG_TABLE_BITS: 23

TAG_BITS: 17

These parameters provided the best balance between prediction accuracy and (somewhat) practical storage requirements. My average number of misses per thousand instructions (MPKI) was 5.013.