Joachim Galil
Prof Tullsen
CSE 240A
11/12/2025

Branch Predictor Report

Overview:

This predictor implemented in this project is a hybrid, tournament predictor featuring a gshare and local predictor. The predictor is a modified version of the Alpha 21264 processor, where the global predictor of the original Alpha 21264 predictor is replaced with a gshare predictor to more accurately catch global history and PC-specific correlations more effectively.

Architecture:

High Level:

Three main components:

1. GShare
    a. Standard gshare predictor
2. Local predictor
    a. 2 bit saturating counters using PC as index
3. Choice predictor
    a. Saturating 2 bit counters to decide which predictor is more accurate for the specific case

This design allows the predictor to use both local and global branch characteristics to make predictions and the choice predictor optimizes this further by choosing the prediction from the historically most accurate predictor.

Component 1: GShare Predictor

Global History Register: 17 bit shift register, tracking last 17 branches (c_ghistory)

Pattern history table of 2^17 entries of saturating 2b counters (c_bht_gshare)

Gshare is highly effective at capturing correlated branch patterns where the outcome of one branch depends on outcomes of previous branches. The XOR operation distributes branches across the PHT while maintaining correlation information. Using 17 bits of global history allows the predictor to capture moderately long sequences of branch outcomes, which is particularly effective for nested loops and correlated conditionals.

Why GShare vs the regular tournament predictor's global predictor? GShare is more effective in predicting branches than the regular global predictor and suffers less from aliasing. The result is a large performance benefit coming from the reduced misprediction rate.

Component 2: Local History Predictor

The local predictor consists of two tables: a local history table that stores per-branch history patterns, and a local pattern table that predicts based on those patterns.
Implementation Details:

Local History Table: 2^13 entries of 10-bit history registers (c_bht_local)
PC Indexing: Lower 11 bits of PC used to index local history table
Local Pattern Table: 2^13 entries of 3-bit saturating counters (c_local_prediction_table)
Pattern Indexing: 13-bit local history used to index pattern table
Counter Range: 0-7, with threshold at 4 (predict taken if counter > 3)

Reasoning:
Local predictors excel at capturing repetitive, self-correlated patterns within individual branches. Many branches have strong correlations, so they are repetitive and through the large local history table, many branch patterns are stored allowing it to recognize these sequences and more accurately predict the branch outcome.

Component 3: Choice Predictor

Choice Table: 2^17 entries of 2-bit saturating counters (c_choice_predictor)
Index Calculation: choice_index = (PC_lower_11_bits ^ c_ghistory) & 0x7FFF
Selection Logic: If counter >= 2, choose local predictor, else choose gshare
Training: Only update when predictors disagree, increment toward correct predictor up for local, down for gshare

Reasoning:
Certain branches are correlated strongly with other branches and some aren't, using both local and gshare predictors allows for choosing the optimal predictor for each case.

Performance: in Misses Per Kilo Instruction(MPKI)

Trace	GShare	Tournament	Custom
U1_Blender	33.65	29.63	29.13
U2_Leela	101.73	97.50	80.87
U3_GCC	19.61	16.11	10.92
U4_Cam4	10.03	6.49	6.56
Average	41.26	37.43	31.87

Notes: Performance across the board is about 2% flat better on gradescope than when run on IENG6. I am not sure why this is the case.