

Branch Prediction Strategies

- ▶ Predict branches based on static information about branch opcodes and directions (1, 1a, 3)
- ▶ Predict the same outcome as the previous execution (2, 4, 5, 6)
- ▶ Use a counter which is incremented/decremented each time the branch is taken/not taken (7)
- ▶ Proposed but not tested: for low-confidence predictions, limit the progression of instructions through the pipeline

What Makes Branch Prediction Possible?

- ▶ There are correlations between outcomes of different branches
- ▶ Global prediction: predict the outcome of branch x based on the outcomes of the previous n branches
- ▶ There are also correlations between a branch's previous and current outcomes (e.g., loop conditions)
- ▶ Per-address prediction: predict the outcome of branch x based on the outcomes of previous executions of x
- ▶ Both types of predictors fail to exploit some of the correlation

Branch Prediction with Perceptrons

- ▶ Use a simple neural network to learn the branch outcome as a function of global history
- ▶ Can outperform gshare because the perceptron learns to ignore parts of the history that are not useful for prediction

Questions

- ▶ How much of an effect do small accuracy improvements have (such as going from 90% to 91% accuracy)?
- ▶ Do modern branch predictors actually take advantage of confidence information?
- ▶ Is the perceptron branch predictor actually practical for use in modern processors?
- ▶ What other types of machine learning might be applicable to branch prediction?