Sidharth Alok Girdhar
305819937

The idea behind my design was adapted from the initial gshare implementation in the skeleton code, as well as the methodology outlined in the paper *Dynamic Branch Prediction with Perceptrons* by Daniel A. Jimenez and Calvin Lin.

The changes to the initial gshare implementation can be summarized by tuning the table size and history length. Moreover, instead of the gshare table indices storing a bimodal predictor, instead a trimodal predictor is stored (done simply by shifting one more bit to get the prediction bit, as well as increasing the maximum value from 3 to 7).

In the paper, a basic formulation of a perceptron is given, as well as some more helpful information such as a formula for the threshold value on when to update the perceptron weights. The paper also mentions using a hybrid model of both a perceptron and gshare table. The paper's implementation involves some method for choosing between gshare and the perceptron, adapted from the Alpha 21264 RISC processor. For my implementation I took a more simple route. Instead of creating another decision mechanism (for the choice predictor), instead the decision outcome of the gshare table was also evaluated as an input variable to the perceptron ANN with its own weight.

The cost of a decision mechanism is however minimal, as the decision mechanism is a global bimodal counter, meaning only two more bits of information were needed for implementation. It should be noted that the benefits of this choice mechanism were found to be minimal, and the addition of gshare in the ANN provided a far greater impact than the addition of the choice mechanism over just the perceptron.

All values for table sizes were determined by testing out values and taking into consideration the time and memory usage. The total memory usage of my implementation should be around 10 MB.