There are 3 branch predictors(local, gshare, tournament) implemented in the file 's1923864.cpp'. Local branch predictor code begins from 72nd line to 149th line. Gshare branch predictor code begins from 153rd line to 230th line. Tournament branch predictor code begins from 235th line to 308th line.

There are 4 member variable in local branch predictor (UINT64 Entries, int LHRS[128], int *PHTS, int bits). Entries represent the number of entries for pattern history table. LHRS[128] represent local history registers with 128 entries. *PHT represent pattern history table. bits represent the number of LSBs of PC needed to map to the entries of pattern history table. In this code, variable index represents the entry for PHTS and address represents 7LSBs of PC. To implement getPrediction(ADDRINT branchPC), firstly calculate PHTS[index]. Find the 7LSBs of PC and map it to local history registers. When PHTS[index] is greater than 1, return true. Otherwise, return false. To implement train(ADDRINT branchPC, bool branchWasTaken), firstly calculate PHTS[index]. Then update PHTS[index] and shift LHRS[address] to right by 1 bit. If branch is taken, plus 1(otherwise plus 0) to the left most bit of LHRS[address]. If the value is greater than 0 and branch is not taken, update current value of PHTS[index] by minus 1. Otherwise, If the value is less than 3 and branch is taken, update current value of PHTS[index] by plus 1. The member variable in gshare branch predictor is same as local branch predictor except that int LHRS[128] is changed to int GHRS, which means there is only a single global register. The way to find the entry(index) for gshare is to do xor operation between GHRS and n-LSBs of PC(n is member variable bits). getPrediction(ADDRINT branchPC) is same as in local branch predictor (PHTS[index] > 1, return true. Otherwise, return false). The update policy of train(ADDRINT branchPC, bool branchWasTaken) in gshare is also same as in local branch predictor. There are 5 member variables in Tournament Branch Predictor (UINT64 Entries, int *PHTS, int bits, LocalBranchPredictor *lbp, GshareBranchPredictor *gbp). lbp is Local Branch Predictor and gbp is Gshare Branch Predictor. The entry(index) for PHTS is n-LSBs of PC. The realization of getPrediction(ADDRINT branchPC) is that if PHTS[index] is greater than 1, return the result of gshare branch predictor. Otherwise, return the result of local branch predictor. For the implementation of train(ADDRINT branchPC, bool branchWasTaken), if gshare result is right and local result is wrong and PHTS[index]<3, update current value of PHTS[index] by plus 1. If gshare result is wrong and local result is right and PHTS[index]>0, update current value of PHTS[index] by minus 1. Otherwise, do nothing. Finally, update lbp's LHRS and gbp's GHRS by recall their own train function.

For local, gshare and tournament predictor, larger PHT size will have better prediction accuracy because larger size reduces the conflict of PC(for local), larger size gives more pattern information at different global pattern(for gshare) and tournament is consist of local and gashare predictor(for tournament). Advantages of Local branch predictor are Good for biased branches and No interference. Disadvantage of Local branch predictor is that it can't discern patterns. Advantages of gshare branch

predictor is that it leverages correlated branches and identifies patterns. Disadvantage of gshare branch predictor is that it can't always take advantage of biased branches. For Tournament predictor, it is good for both biased and unbiased branches and it can be shown from following pictures that Tournament predictor always has highest prediction accuracy.