Cesar Marin Margaillan

CSE 420

Assignment 3

Section I. Branch Prediction Algorithms

In this lab, we used 3 different branch predictors:

Local 2-bit(Included)

The local 2 bit branch predictor works by having a local history table. Every entry in this table contains 2 bits which represent the decision to be made for that branch (take or not take). The branch address is used to hash the index for the table. The decision works as a 4 state FSM (2 bits to represent) and changes the prediction every time it commits 2 mispredictions in a row.

Tournament(Included)

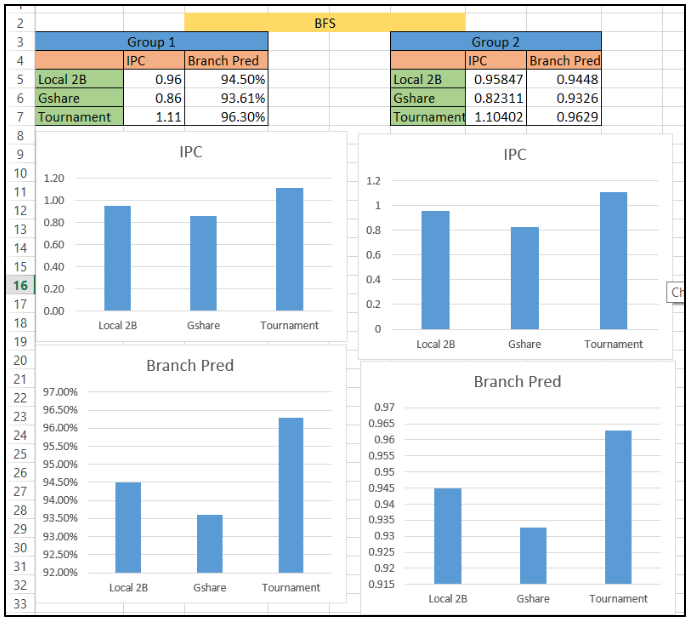
Tournament branch predictor works by having 2 separate predictors to predict a take/not take branch, and then having a third predictor that predict which one of the two branch predictors to use. One of the branch predictors in tournament is the Local 2-bit predictor explained above. The other is a global history predictor. Instead of having an entry for every branch address, the global predictor takes into account all branch history, no matter which branch it was. This can be done with a saturating counter or a shift register. The third is a choice predictor, which can be done with a saturating counter to predictor which one of the two (local or global) to use.

GShare(Added)

GShare also uses global history, but in a different way. GShare also has a 2-bit local history table. In this case, the global history is stored in a shift register of taken/not taken history (no matter the branch address). This shift register is then combined (in this lab, it is XOR) with the branch address to hash into the local table. The local table works the same as above, with a 4 state FSM to decide a take/not take branch prediction.

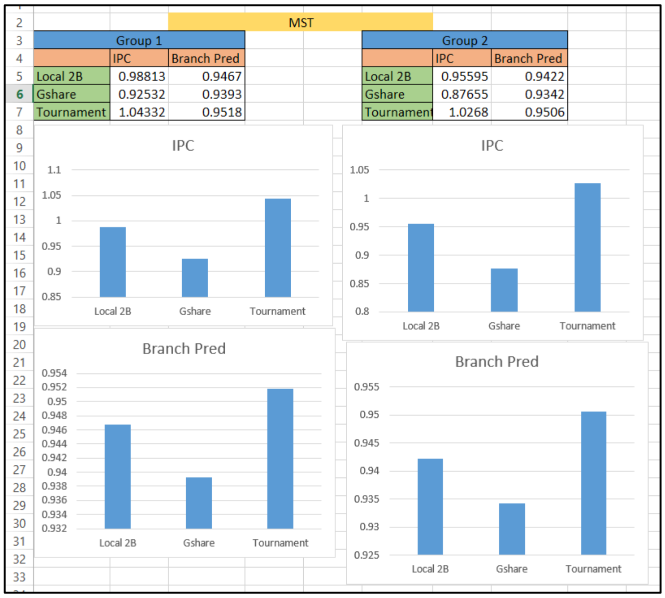
Section II. Data

In this assignment, then, we used gem5 to test each of these 3 different branch predictors across all 4 benchmarks (BFS, MST, SHA, QUEENS). We performed this test for the 3 predictors under 2 groups. In group 1 we kept the total sizes of the predictors at around 32K. In Group 2 we lowered the size of all predictor to around 8K.



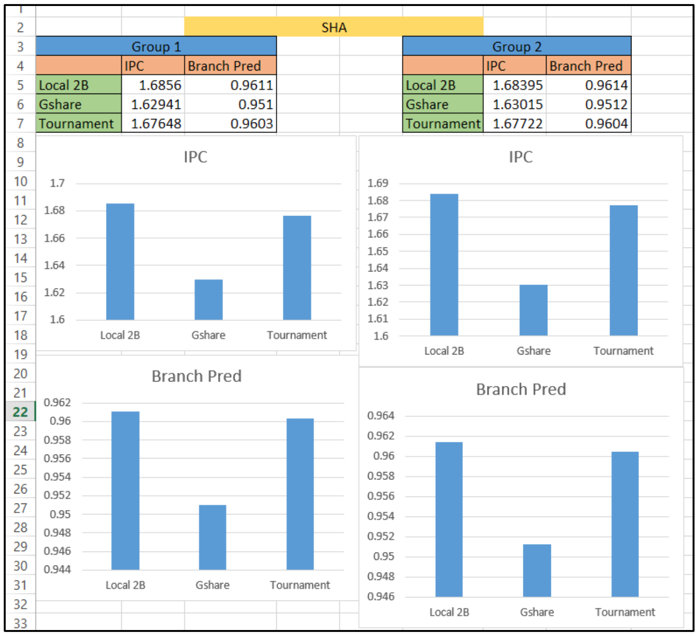
BFS

	Group 1			Group 2	
	IPC	Branch Pred		IPC	Branch Pred
Local 2B	0.96	94.50%	Local 2B	0.95847	0.9448
Gshare	0.86	93.61%	Gshare	0.82311	0.9326
Tournament	1.11	96.30%	Tournament	1.10402	0.9629



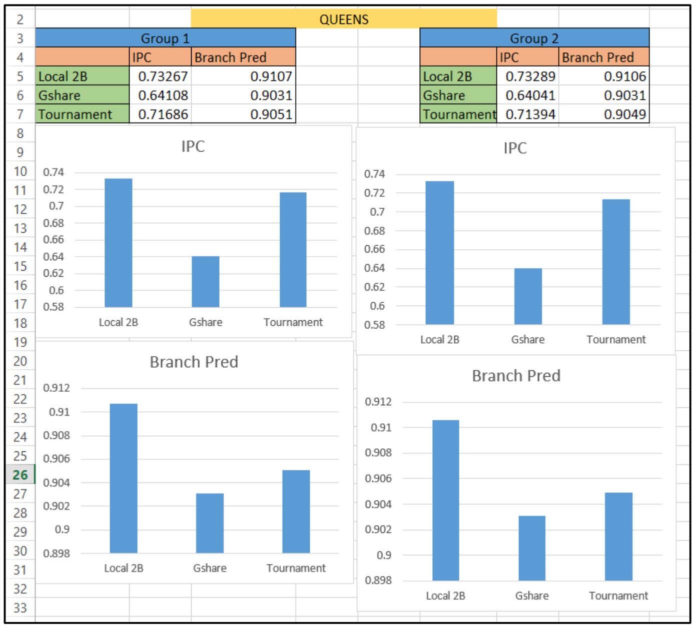
MST

	Group 1			Group 2	
	IPC	Branch Pred		IPC	Branch Pred
Local 2B	0.98813	0.9467	Local 2B	0.95595	0.9422
Gshare	0.92532	0.9393	Gshare	0.87655	0.9342
Tournament	1.04332	0.9518	Tournament	1.0268	0.9506



SHA

	Group 1			Group 2	
	IPC	Branch Pred		IPC	Branch Pred
Local 2B	1.6856	0.9611	Local 2B	1.68395	0.9614
Gshare	1.62941	0.951	Gshare	1.63015	0.9512
Tournament	1.67648	0.9603	Tournament	1.67722	0.9604



QUEENS

	Group 1			Group 2	
	IPC	Branch Pred		IPC	Branch Pred
Local 2B	0.73267	0.9107	Local 2B	0.73289	0.9106
Gshare	0.64108	0.9031	Gshare	0.64041	0.9031
Tournament	0.71686	0.9051	Tournament	0.71394	0.9049

Section III. Observations

As seen above in the data, Gshare turned out to be the worst in all 4 workloads, under both groups.

For both BFS and MST, tournament was better in both IPC and Branch Prediction accuracy than both 2bit and gshare. 2bit followed closely, and gshare was more separated from these two at the bottom. For BFS, Ghsare was pretty close to both tournament and 2bit in IPC, but suffered more when it came to prediction accuracy. In these two benchmarks there didn't seem to be much difference between group 1 and group 2, so the results did not seem to be affected much by size.

For both SHA and QUEENS benchmarks, 2bit took the lead in both IPC and branch prediction accuracy. Tournament followed very closely, and although in numbers gshare was really close, when it comes to scale and standard deviations, it was farther down than 2bit and tournament predictors. Gshare suffered equally in IPC and branch prediction. Again under both groups 1 and 2 there did not appear to be much difference, and the number where actually almost identical, so again it didn't seem that size affected the predictors much. For QUEEENS, tournament also seemed to suffer in branch prediction accuracy, coming down be closer to gshare while 2bit local stayed up.

I am guessing that Gshare had trouble because of the hashing algorithm into the local history table, maybe a different hashing algorithm other than XOR and Modulus would have worked better for these four benchmarks.