Assignment 2

Computer Architecture

Nidhi Hegde (180472)

Part A:

Misprediction Fraction (FCB / BCB / Overall)							
Prediction Technique	**perl**	**bzip**	**mcf**	**gcc**	**soplex**	**hammer**	**xalanc**
FNBT	0.376 / 0.562 / 0.413	0.31 / 0.619 / 0.469	0.357 / 0.282 / 0.319	0.319 / 0.045 / 0.104	0.199 / 0.155 / 0.17	0.766 / 0.007 / 0.639	0.083 / 0.049 / 0.075
Bimodal	0.099 / 0.079 / 0.095	0.108 / 0.092 / 0.1	0.163 / 0.198 / 0.18	0.126 / 0.02 / 0.043	0.009 / 0.067 / 0.048	0.102 / 0.004 / 0.086	0.039 / 0.021 / 0.034
SAg	0.037 / 0.033 / 0.036	0.114 / 0.089 / 0.101	0.151 / 0.11 / 0.13	0.076 / 0.013 / 0.027	0.007 / 0.056 / 0.04	0.109 / 0.005 / 0.091	0.021 / 0.011 / 0.018
GAg	0.124 / 0.091 / 0.117	0.148 / 0.104 / 0.125	0.092 / 0.093 / 0.093	0.12 / 0.019 / 0.04	0.009 / 0.052 / 0.038	0.136 / 0.026 / 0.118	0.049 / 0.031 / 0.044
gshare	0.103 / 0.097 / 0.102	0.121 / 0.105 / 0.113	0.103 / 0.101 / 0.102	0.119 / 0.017 / 0.039	0.013 / 0.053 / 0.04	0.118 / 0.027 / 0.103	0.042 / 0.029 / 0.038
SAg and GAg Hybrid	0.032 / 0.027 / 0.031	0.108 / 0.084 / 0.096	0.089 / 0.086 / 0.088	0.058 / 0.01 / 0.02	0.007 / 0.05 / 0.036	0.102 / 0.008 / 0.086	0.015 / 0.01 / 0.014
SAg, GAg, and gshare Hybrid (Majority)	0.051 / 0.048 / 0.05	0.102 / 0.085 / 0.093	0.088 / 0.086 / 0.087	0.071 / 0.012 / 0.025	0.008 / 0.052 / 0.038	0.103 / 0.006 / 0.087	0.022 / 0.016 / 0.021
SAg, GAg, and gshare Hybrid (Tournament)	0.029 / 0.024 / 0.028	0.101 / 0.082 / 0.091	0.085 / 0.083 / 0.084	0.051 / 0.009 / 0.018	0.007 / 0.05 / 0.036	0.099 / 0.004 / 0.084	0.014 / 0.009 / 0.013

**Best overall performance for each benchmark highlighted in red*

Observations:

We can see that the **FNBT predicto**r performs the worst for all the benchmarks, while the **Tournament Predictor (SAg, Gag and gshare)** performs the best, followed by **SAg and GAg Hybrid predictor** and **(SAg, Gag and gshare) Majority predictor** for most of the benchmarks.

Explanation:

This is because FNBT offers a static prediction, giving a prediction 1 for all the backward branches. It may perform well only in the case of loops, but not up to mark in other cases.

The tournament predictor on the other hand captures both the global and local patterns in the binary and tries to learn the best prediction among Sag, Gag and gshare over time. Hence, we see that such techniques using a meta-predictor tend to give better results over the naïve majority selection techniques, in case of Hybrid predictors.

Part B:

Misprediction Fraction / Miss Rate	BTB Type	BTB (Indexed with PC)	BTB (Indexed with PC and GHR Hash)
	perl	**0.3523009228 / 0.0034048852**	**0.1057537398 / 0.0206169801**
	bzip	**0.4829537562 / 0.0000745094**	**0.4752515009 / 0.0002702546**
	mcf	**0.006135006 / 0.0000007168**	**0.0040994812 / 0.0000083622**
	gcc	**0.35513708 / 0.0281566251**	**0.1226970983 / 0.0376688392**
	soplex	**0.0000535151 / 0.0000148829**	**0.0001037054 / 0.0000530402**
	hammer	**0.0635823518 / 0.0003176636**	**0.0258300202 / 0.0036233503**
	xalanc	**0.2738973095 / 0.0238351259**	**0.2879870846 / 0.1627983638**

Observations:

We see that **BTB (Indexed with PC**) offers a lower miss rate for all the benchmarks, but **BTB (Indexed with PC and GHR Hash)** consistently offers better misprediction fraction (except for xal where both are similar).

Explanation:

Every time a branch instruction is encountered, it misses in the first turn after which its target is stored to the BTB (if the branch is taken). Thus, the number of misses is proportional to the number of unique branch instructions in case of BTB indexed with PC.

A higher miss rate in case of BTB indexed with the Hash of PC and GHR is because the probability of a given instruction falling into a particular bucket reduces on XORing the PC with GHR. Thus, every time that instruction executes, it may produce a different hash value depending on the GHR value.

Secondly, in case of BTB indexed with the PC and GHR, since the global execution path is taken into account, it offers better prediction than a naïve BTB indexed with PC.