1. Introduction:

A branch predictor is a digital circuit that tries to guess which way a branch (e.g. an if then else statement) will go before this is known definitively. The purpose of the branch predictor is to improve the flow in the Instruction pipeline. Branch predictors play a critical role in achieving high effective performance in many modern pipelined microprocessor architectures.

Types of branch predictors considered in this assignment are as follows:

a) 2-bit Local predictor :

- This predictor changes prediction only on two successive mis-predictions. Two bits are maintained in the prediction buffer and there are four different states. Two states corresponding to a taken state and two corresponding to not taken state.
- Advantage of this approach is that a few typical branches will not influence the prediction (a better measure of the common case). Only when we have two successive mis-predictions, we will switch prediction. This is especially useful when multiple branches share the same counter (some bits of the branch PC are used to index into the branch predictor). This can be easily extended to N-bits. Suppose we have three bits, then there are 8 possible states and only when there are three successive mis-predictions, the prediction will be changed. However, studies have shown that a 2-bit predictor itself is accurate enough.

b) Tournament predictor:

- This uses the concept of - Predicting the predictor and hopes to select the right predictor for the right branch. There are two different predictors maintained, one based on global information and one based on local information, and the option of the predictor is based on a selection strategy. For example, the local predictor can be used and every time it commits a mistake, the prediction can be changed to the global predictor. Otherwise, the switch can be made only when there are two successive mis-predictions. Tournament predictors using » 30K bits are used in processors like the Power5 and Pentium 4. They are able to achieve better accuracy at medium sizes (8K – 32K bits) and also make use of very large number of prediction bits very effectively. They are the most popular form of multilevel branch predictors which use several levels of branch-prediction tables together with an algorithm for choosing among the multiple predictors.

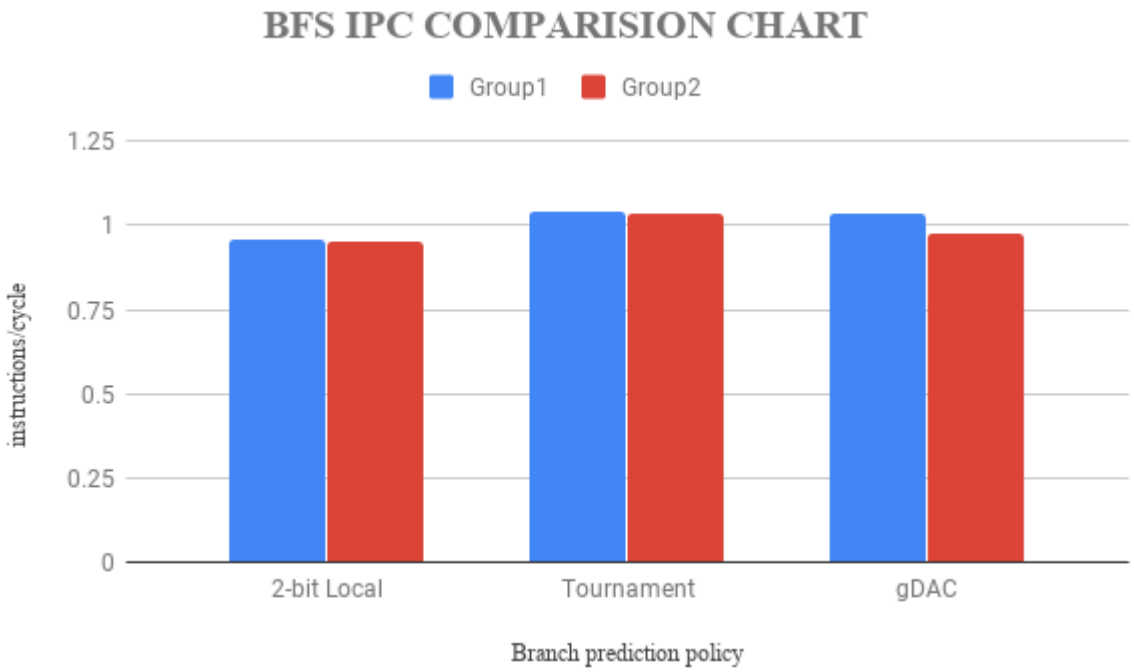
c) gDAC predictor:

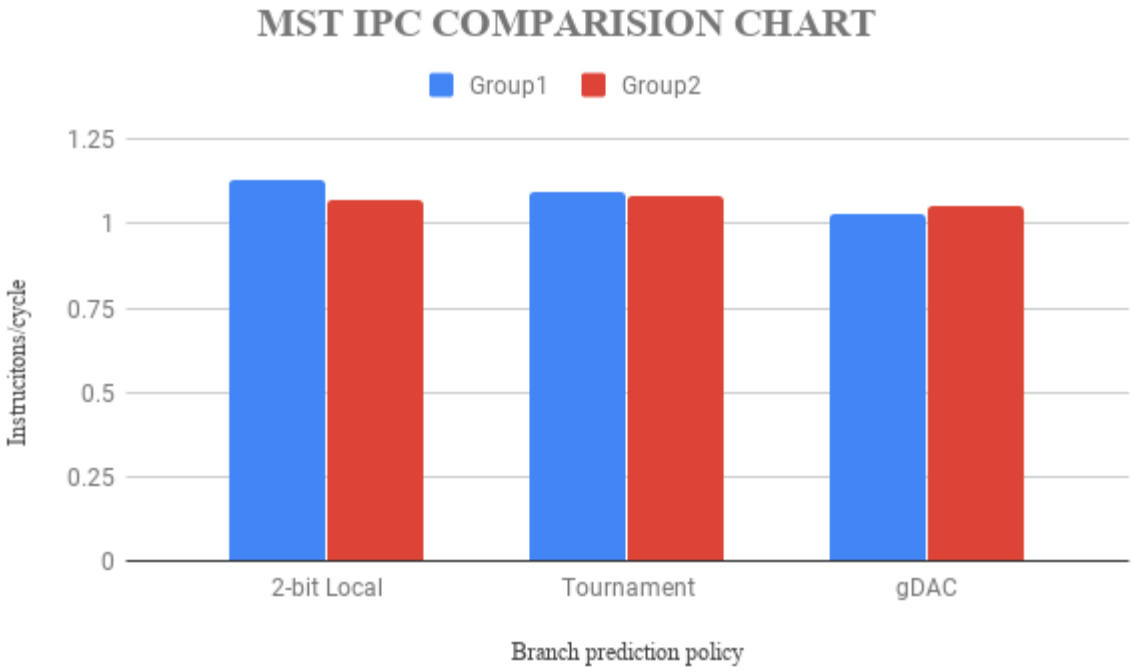
- global-history Divide-and-Conquer (gDAC) branch predictor combines the ideas of ahead-pipelining, history segmentation, and prediction fusion.
- In this, We divide the long global branch history register into non-overlapping segments. Each segment provides a branch history input to a PHT-based predictor that provides a branch prediction for that segment. A final root predictor takes all of the per-segment predictions as part of its input vector and computes the final prediction.
- In our case, gDAC root predictor is a simple fusion predictor. We hash the most recent global history with the branch address, and then concatenate the per-segment predictions to form a final index. The index selects a 4-bit counter from a standard PHT structure where the most significant bit of the counter determines the final prediction.
- To reduce the effects of destructive interference in the segment predictors, gDAC uses a Bi-Mode organization instead of conventional gshare style predictors. A Bi-Mode predictor sorts branches into two tables: one that tracks branches that are usually taken, and a second that tracks branches that are usually not-taken. The idea is that if two branches map to the same entry, but both are usually taken (or both not-taken), then the interference will be neutral. A third table called the choice-PHT tracks the bias of each branch and is used to select one of the two tables.
- Implementation of gDAC in our case : we have 2 segment predictors and a 1 choice table. The choice table is using 1.5 bit counter where 1 bit is taken from direction table and other is taken from hysteresis table. The decision from the 2 segment predictors is concatenated with the branch history table and PC which is used as the index for the fusion table that gives the final prediction.

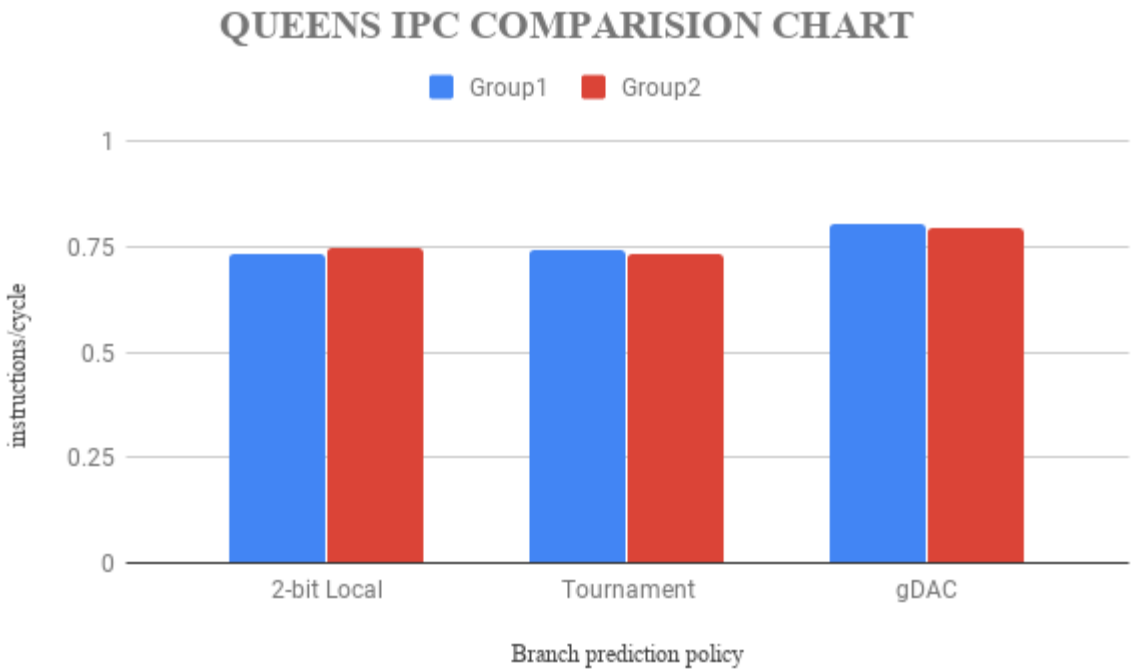
2. Result and Observations:

IPC TABLE

	BFS	MST	queens
Local32k	0.95592	1.128276	0.73506
Local8k	0.952803	1.07232	0.745403
Tournament32k	1.042682	1.092773	0.740442
Tournament8k	1.035277	1.07996	0.735137
gDAC32k	1.032276	1.031924	0.806054
gDAC8k	0.97283	1.053389	0.795748

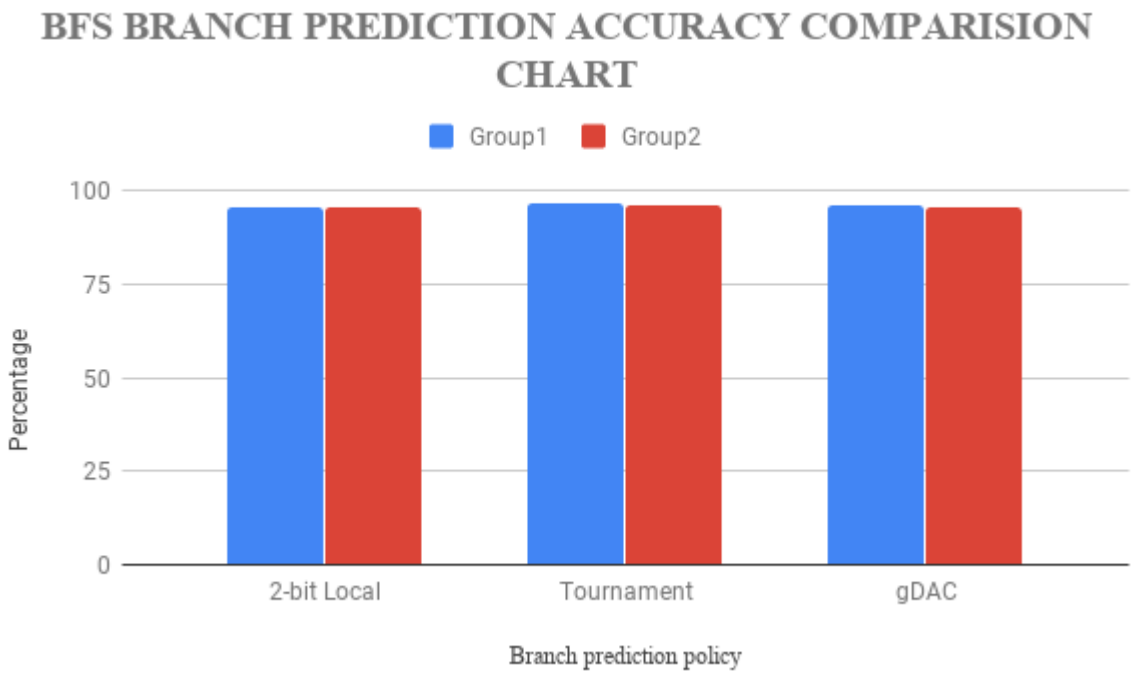

BFS IPC COMPARISION CHART
Group1
Group2
1.25
1
0.75
0.5
0.25
0
instructions/cycle
2-bit Local
Tournament
gDAC
Branch prediction policy

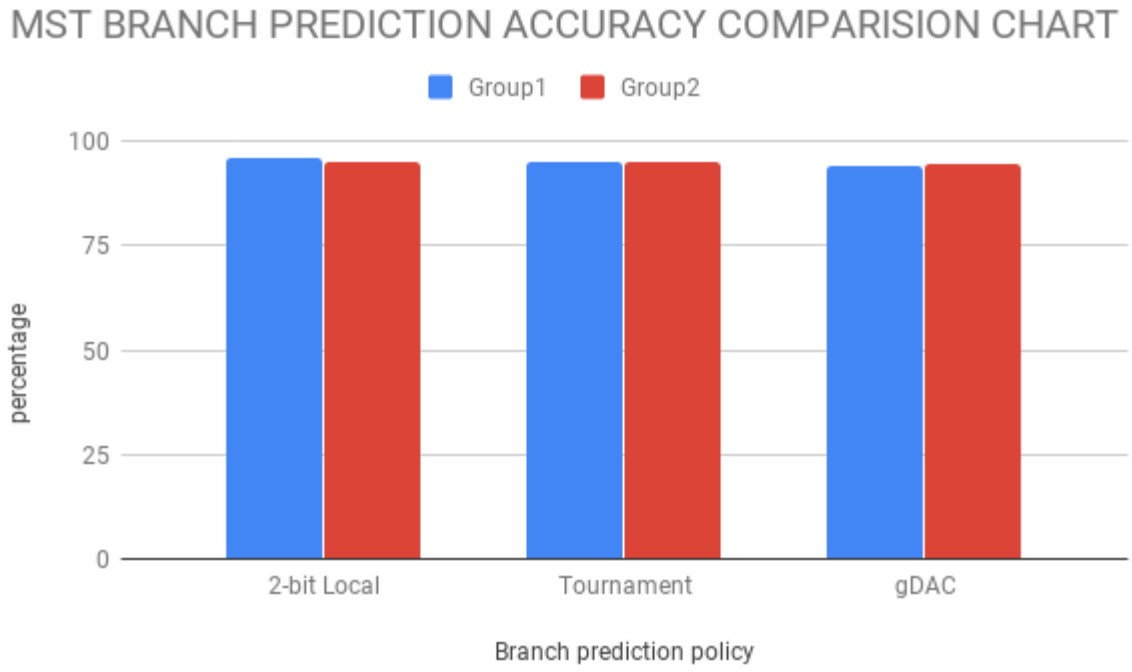

MST IPC COMPARISION CHART
Group1
Group2
1.25
1
0.75
0.5
0.25
0
Instrucitons/cycle
2-bit Local
Tournament
gDAC
Branch prediction policy

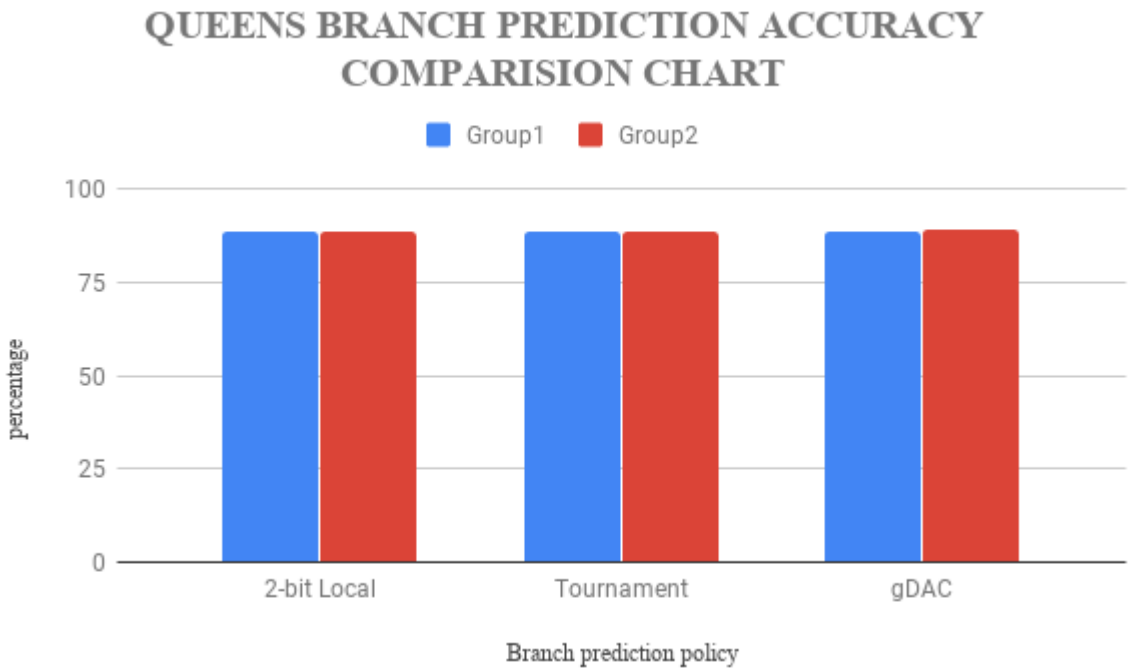


BRANCH PREDICTION ACCURACY TABLE

	BFS	MST	queens
Local32k	0.956593322	0.958365761	0.88609678
Local8k	0.956422548	0.949784268	0.885934423
Tournament32k	0.964639273	0.952663206	0.887288636
Tournament8k	0.963461932	0.952780325	0.886427069
gDAC32k	0.962677788	0.943559623	0.88837714
gDAC8k	0.954634604	0.946915463	0.88917846


BFS BRANCH PREDICTION ACCURACY COMPARISION CHART
Group1
Group2
100
75
50
25
0
Percentage
2-bit Local
Tournament
gDAC
Branch prediction policy


MST BRANCH PREDICTION ACCURACY COMPARISION CHART
Group1
Group2
100
75
50
25
0
percentage
2-bit Local
Tournament
gDAC
Branch prediction policy



3. Inference and Conclusion:

Accuracy results :

- The gDAC predictor provides prediction accuracies that are comparable to the other predictors. At moderate predictor sizes(8KB), gDAC catches up with the tournament predictor. Even at the largest sizes(32KB) considered, the Tournament and 2-bit predictors consistently provides the best prediction accuracy rates. The reasons that tournament predictor provide a better overall accuracy is that it makes use of the full branch address, whereas gDAC does not due to its pipelined organization. Overall, the gDAC predictor achieves raw prediction accuracy that is on par with the other two predictors. The overall performance impact will also be affected by the latency characteristics of the predictors. On a per-benchmark basis, the 32KB gDAC predictor the gDAC predictor performs slightly worse than the Tournament predictor, but only by a very small margin.

IPC results:

- In case of Queen and BST problems, for all predictor sizes (8K and 32KB), gDAC achieves a better IPC rate than other predictors as there will be a good correlation in the problem which helps gDAC. Also, this is due to a combination of the long latency and, at larger hardware budgets, gDAC's better prediction rate. From 32KB, gDAC delivers more

performance than both the predictors. In case of MST problem, tournament predictor has better IPC rate than gDAC, but only by a very small margin. This might be because MST problem has poor correlation between the nodes inside it which removes the advantage of gDAC.