ENEE 646 Project 4

Evaluating the Performance of Static and Dynamic Branch Prediction Schemes

Dishank Upadhyay

Mukul Kulkarni

Objectives

- Explore the effectiveness of branch direction prediction on an actual program.
- Read a representative trace and compute the accuracy rate of different branch direction prediction schemes.
- Provide a comparison for the following prediction schemes.
- Static Prediction -

1. Always Taken
2. Always Not Taken

- Dynamic Prediction

1. Bimodal Predictor with 1-Bit History
2. Bimodal Predictor with 2-Bit Saturating Counters
3. Gshare Predictor
4. Tournament Predictor

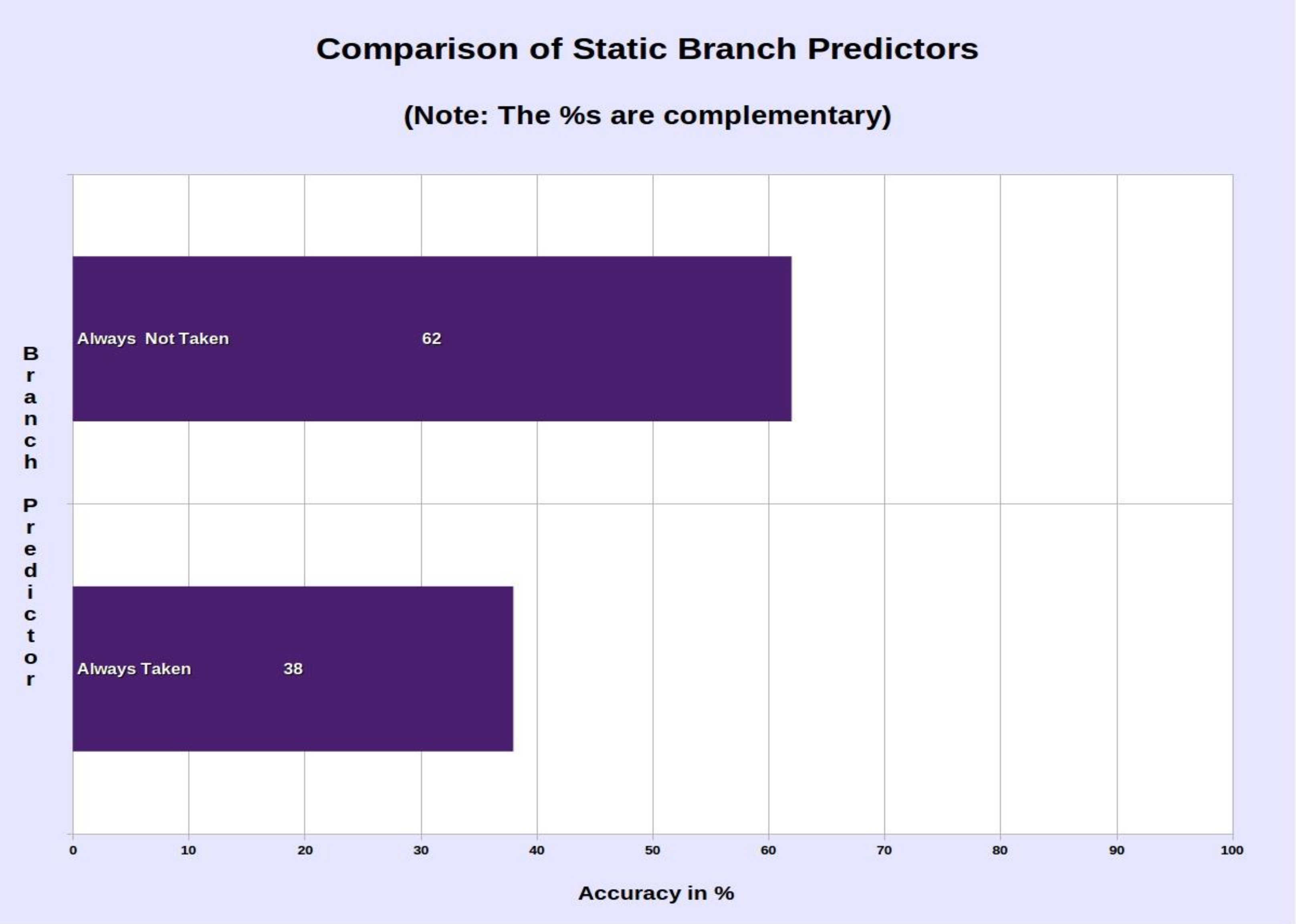
Input Trace file

- Trace of 16 million conditional branches
- These are the conditional branches from an execution of the program GCC (Gnu C Compiler) from the SPECint2000 benchmark suite.
- Each line of the trace file has two fields.

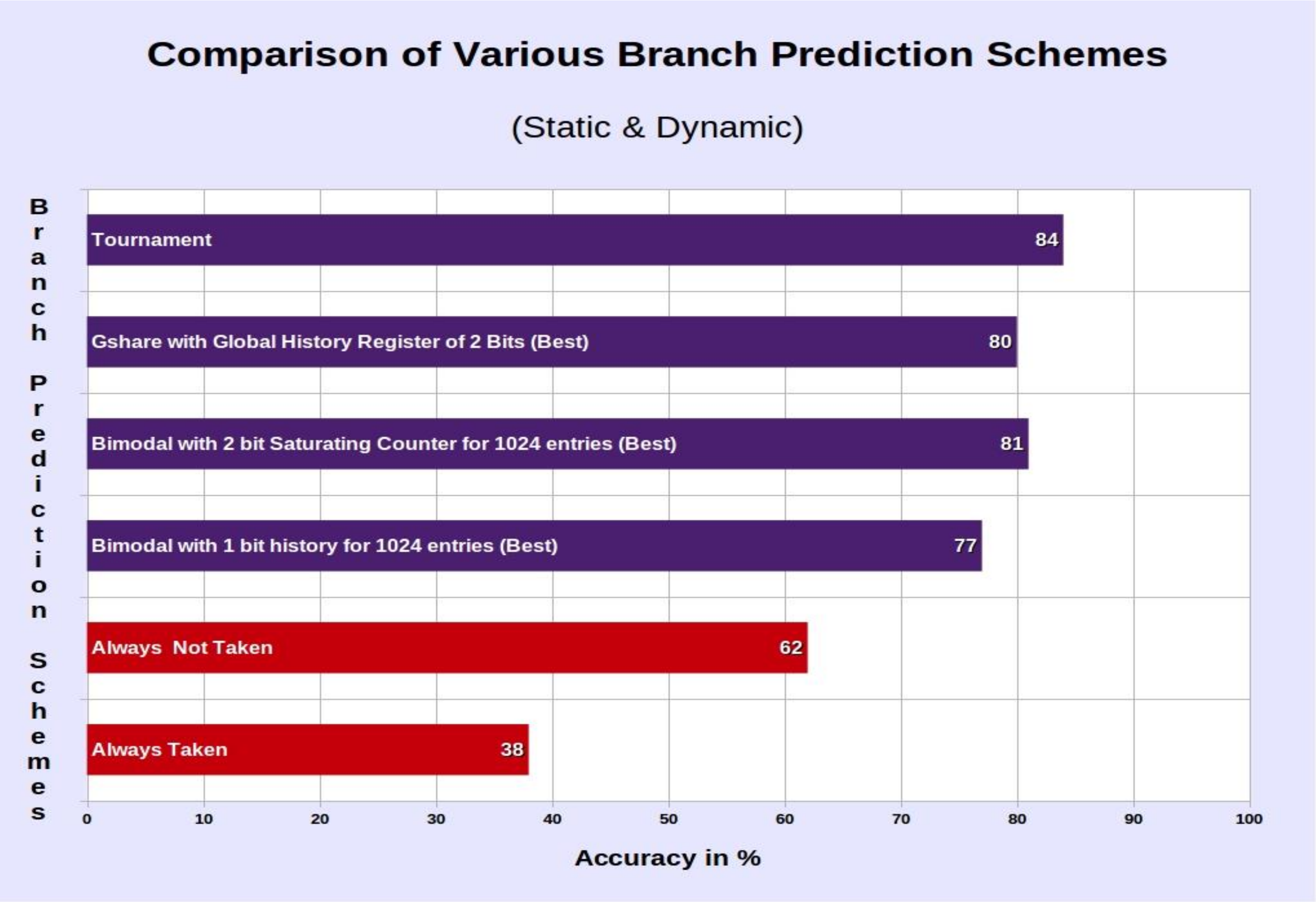
  3086629576 T

  3086629604 N
- The first field is the address of the branch instruction. The second field is the character "T" or "N" for branch taken or not taken
- Obtained using Pin, a binary instrumentation tool (Trace from UPenn CIS Repository)

Results - Static Branch Prediction

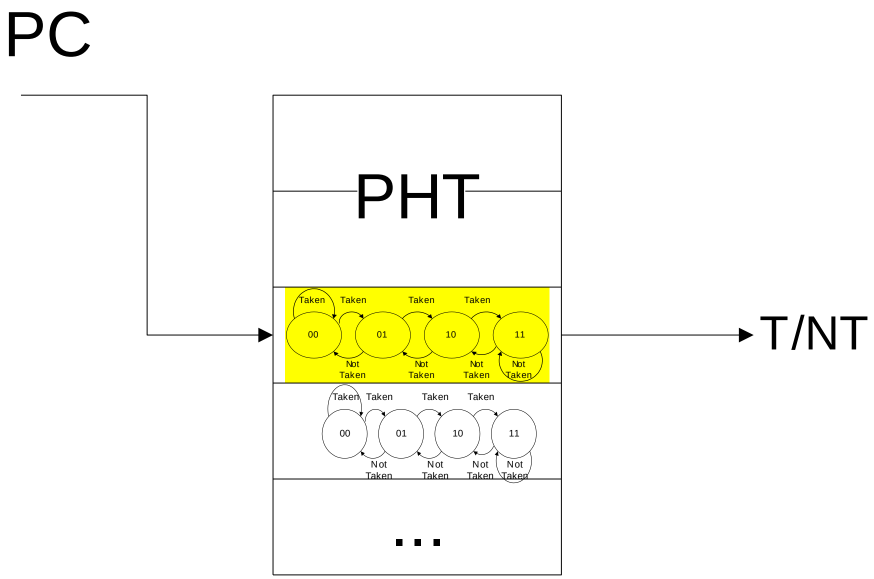


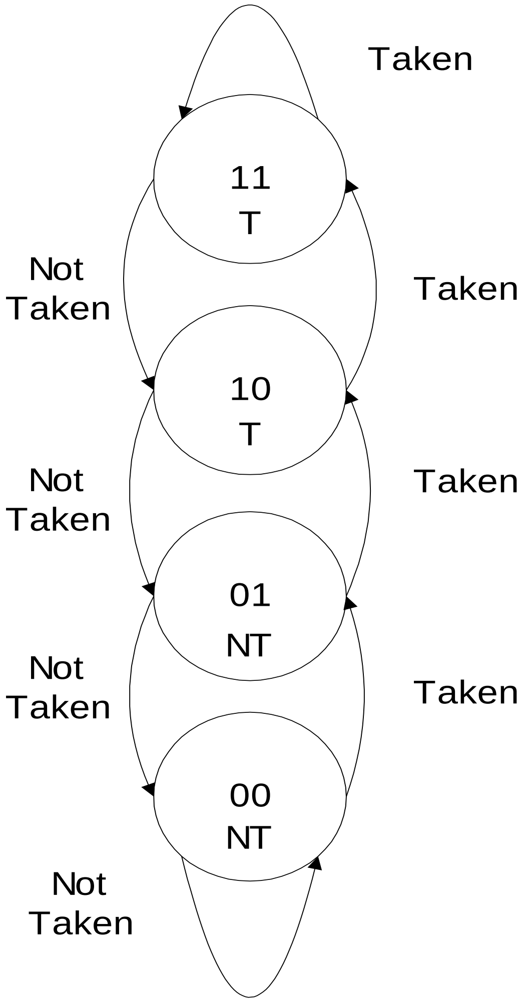
Results – Comparison of Static and Dynamic Schemes



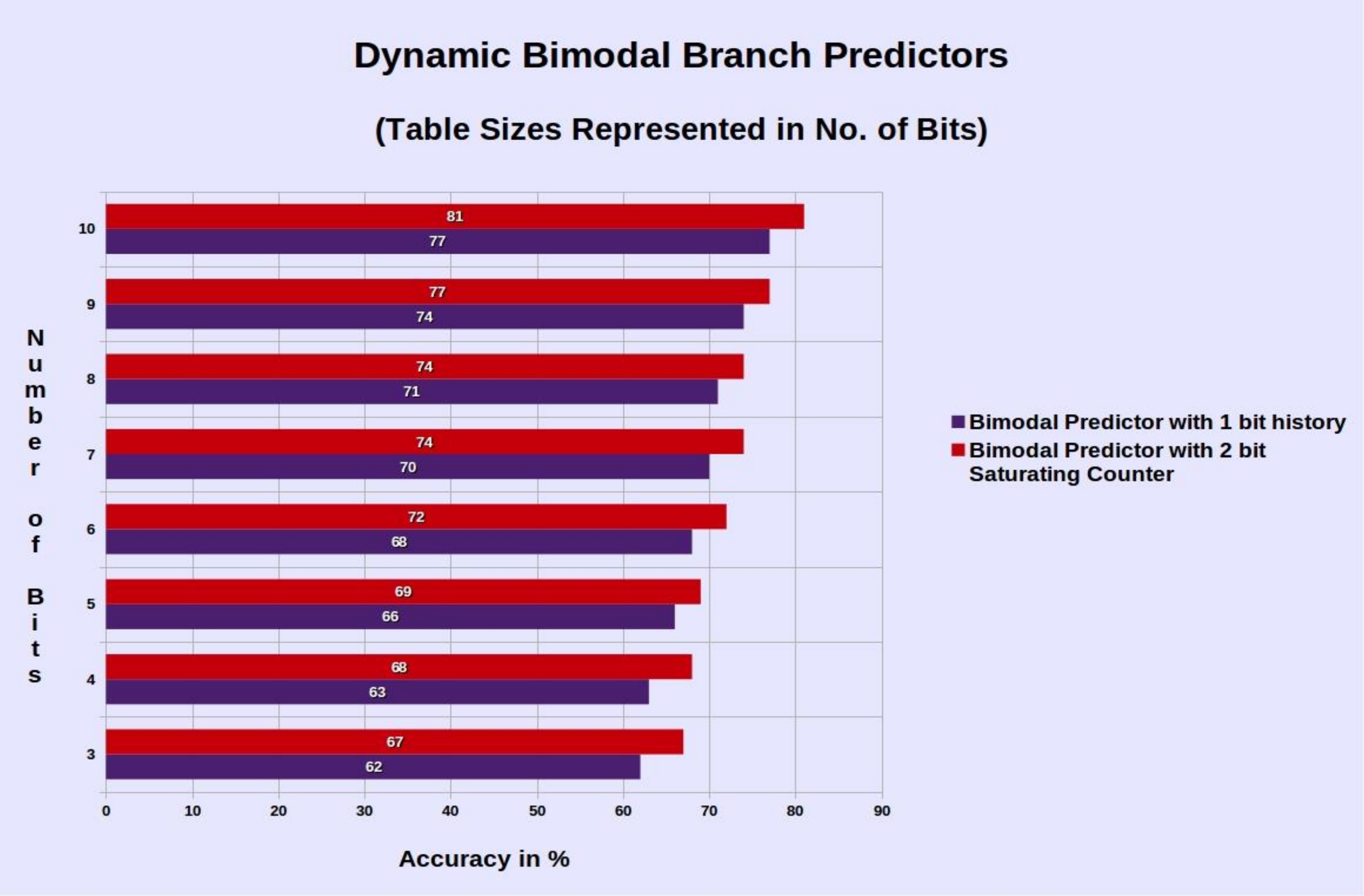
Bimodal Prediction

- Table of 2-bit saturating counters
  - Predict the most common direction

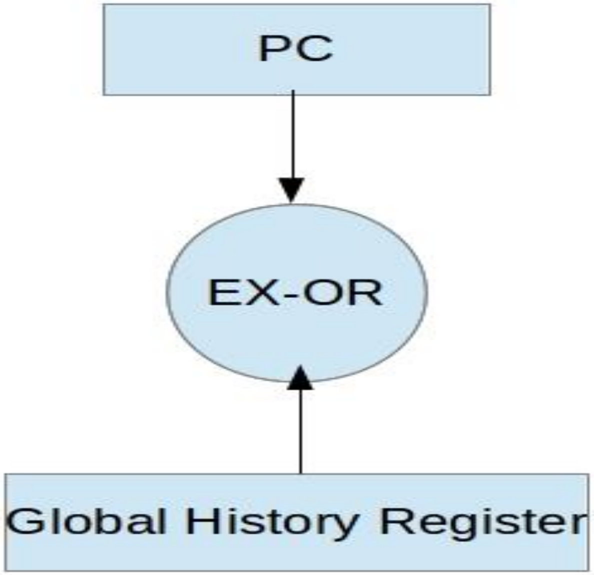




Results – Dynamic Bimodal Predictor



Gshare Predictor



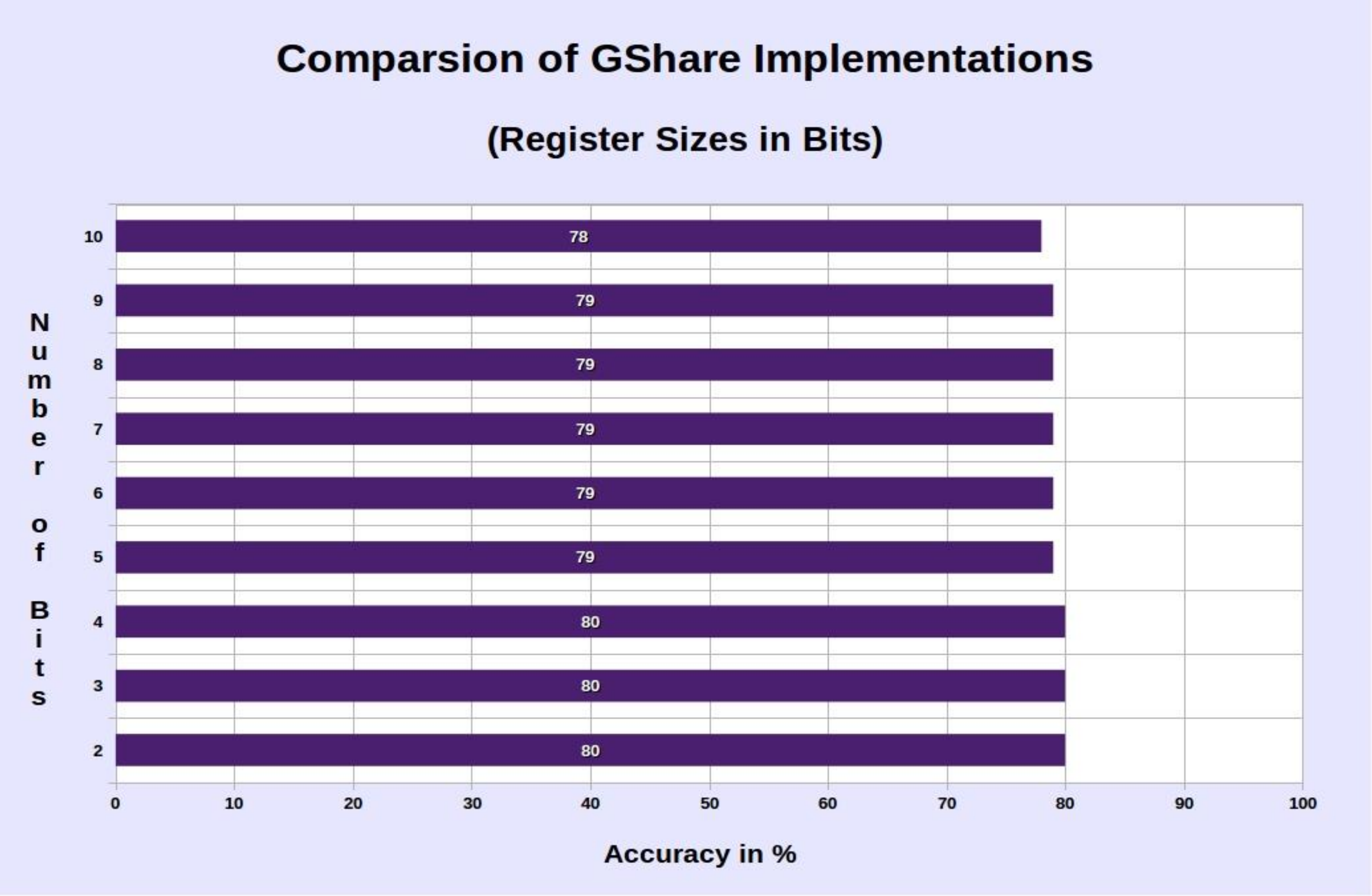
PC

EX-OR

Global History Register

Hashed Address	2-Bit Prediction Value

Results – gshare Predictor



Tournament Predictor

Outlook

- Dynamic Branch Prediction has significantly higher prediction accuracy than static schemes.

- Can design custom hardware for a particular program knowing which scheme is best suited for it.

- **Thank You for your attention !**
- **Questions?**