

# **gselect Implementation and Analysis**

Ibrahim Binmahfood  
Anjela Albaka  
Phil Nevins  
Kunjan Vyas





# Agenda

What is gselect?

Functionality

Implementations on SimpleScalar

Demo

Results

References



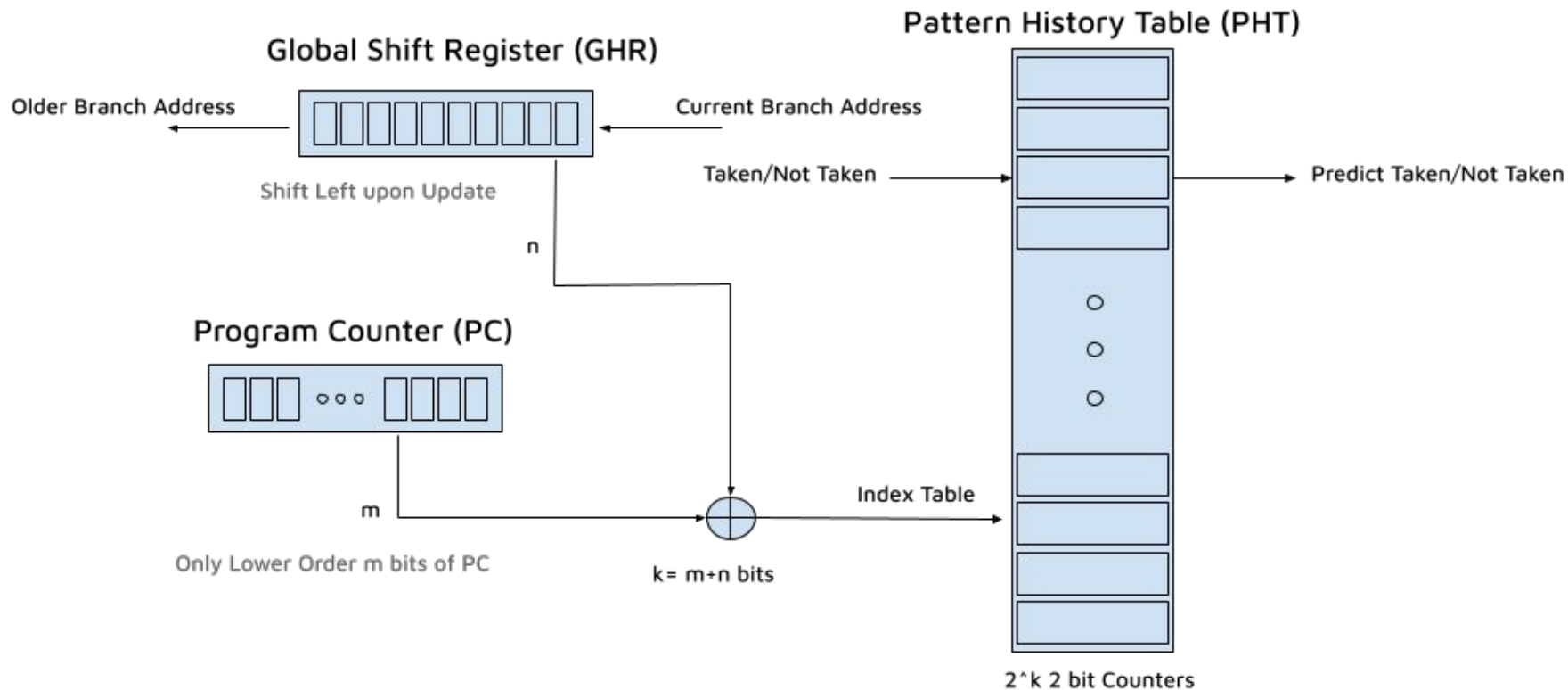


# What is gselect?

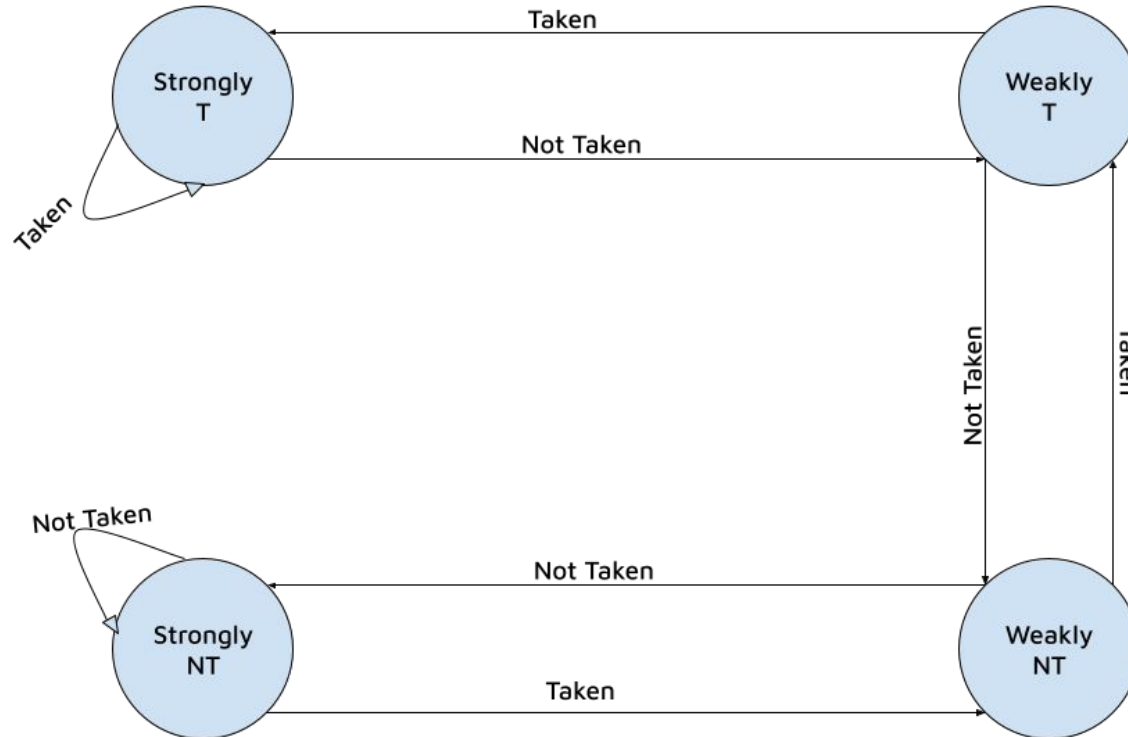
**Gselect** is a type of **branch predictor** that combines global and local information to make predictions about the outcome of a branch. Specifically, it uses a portion of the **Global History Register (GHR)** to index into a table of counters for prediction. Gselect belongs to the class of global branch predictors.



# Functionality



## Functionality - Cont.





# Implementations on SimpleScalar

In **SimpleScalar**, Gselect is implemented as part of the branch prediction unit (**bpred.c**)

## 1. Data Structures

- **Global History Register (GHR)**: A shift register that stores the recent branch outcomes (1 for taken, 0 for not-taken).
- **Prediction Table**: An array of saturating counters (typically 2-bit counters).
- **GHR Length**: Determines how many bits of global history are used.

## 2. Index Calculation:

- Concatenate specific bits of the **PC** and the **GHR** to form the index into the prediction table.

## 3. Prediction

- Fetch the counter value at the computed index.
- Predict "taken" if the counter is above a threshold (e.g., 2 for a 2-bit counter).

## 4. Update

- After the branch resolves, update the saturating counter based on the actual branch outcome (increment if taken, decrement if not-taken).
- Shift the GHR to incorporate the outcome of the resolved branch.



# Demo

Initial Conditions as per McFarling's paper:

- Only 1st 10 million instructions simulated per each benchmark
- All counters set as taken

The given benchmarks with SimpleScalar simulator (**SPEC2000**) has a variety of benchmarks. Some of these benchmarks were used from McFarling's paper.

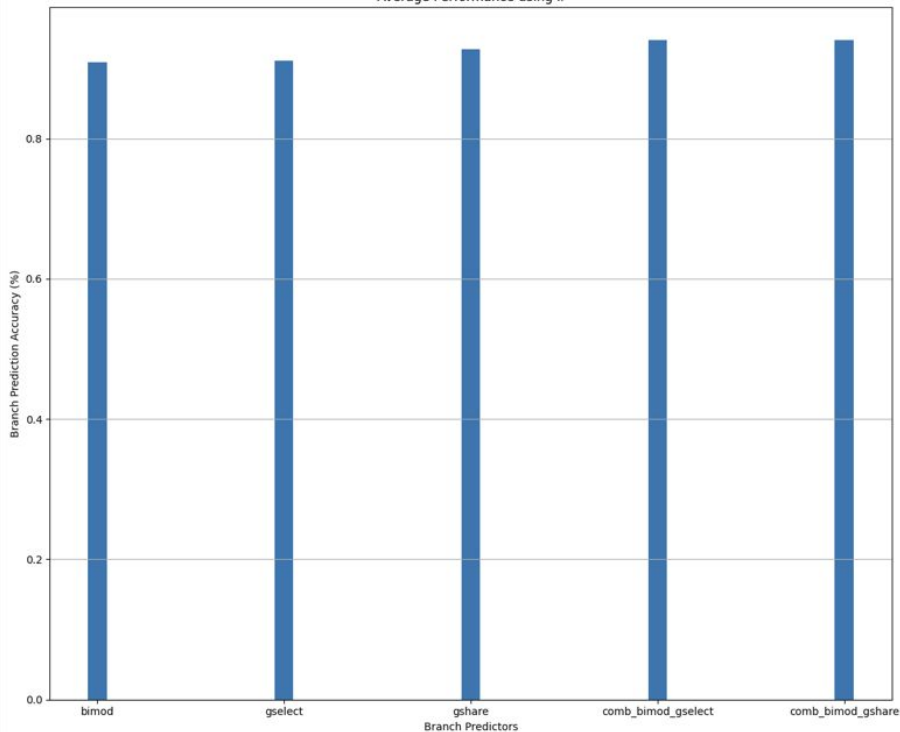
- **gcc**
- **li**
- **tomcatv**
- **fpppp**

However **tomcatv** was not working as expected. The **fpppp** would have been included if it didn't take as long to run. So these two benchmarks will be excluded.

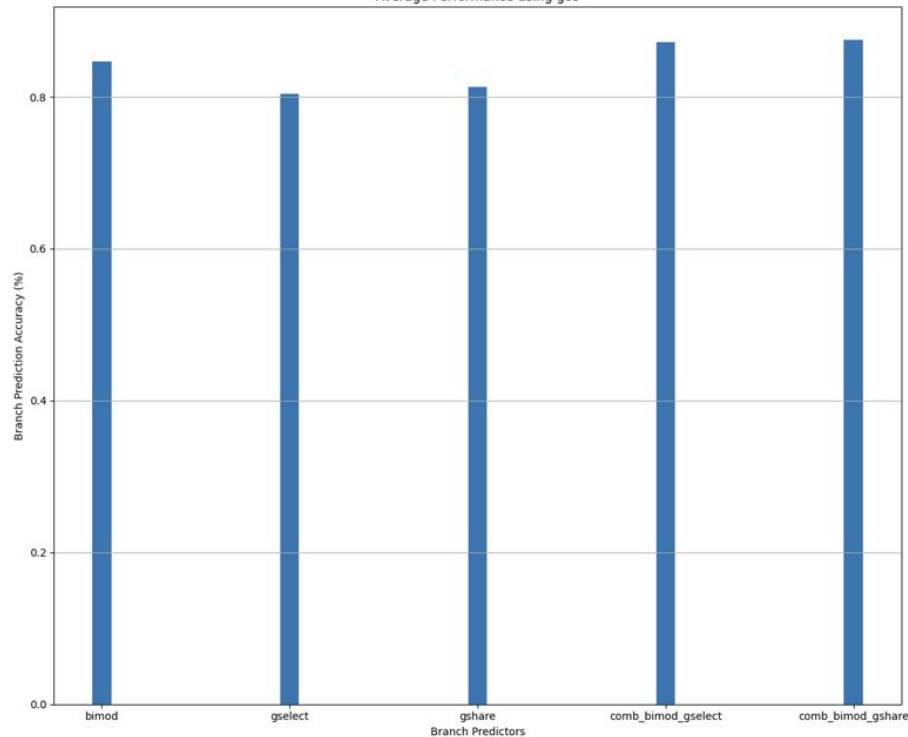


# Results - Accuracy Performance li/gcc - Gselect vs Other Branch Predictors

Average Performance using li



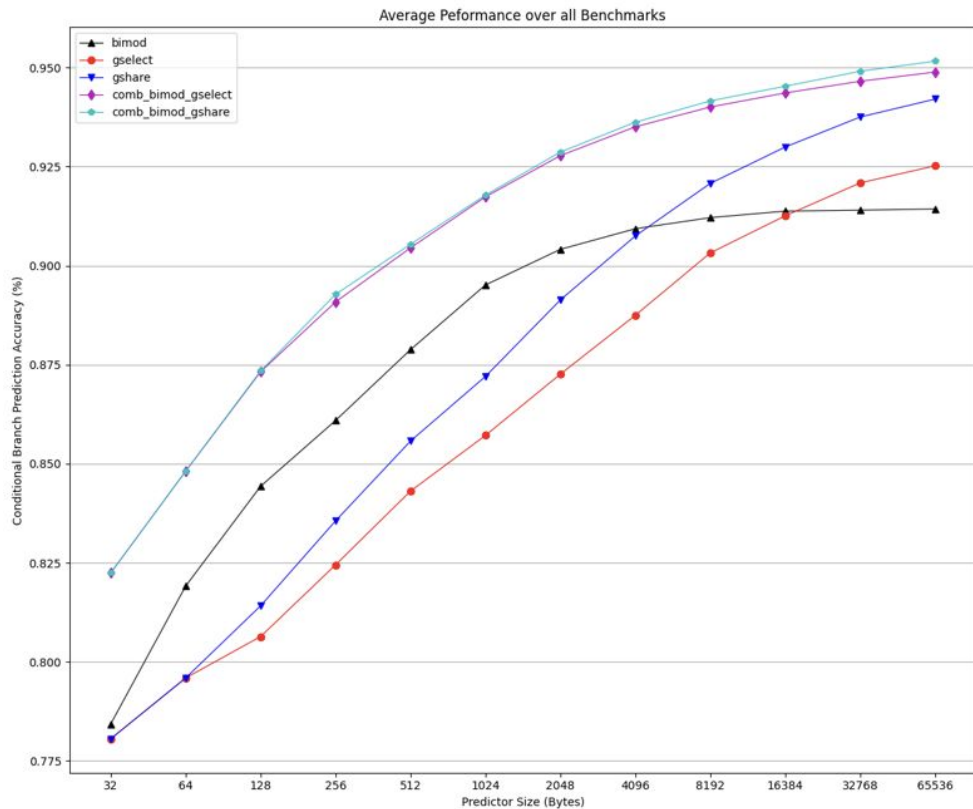
Average Performance using gcc



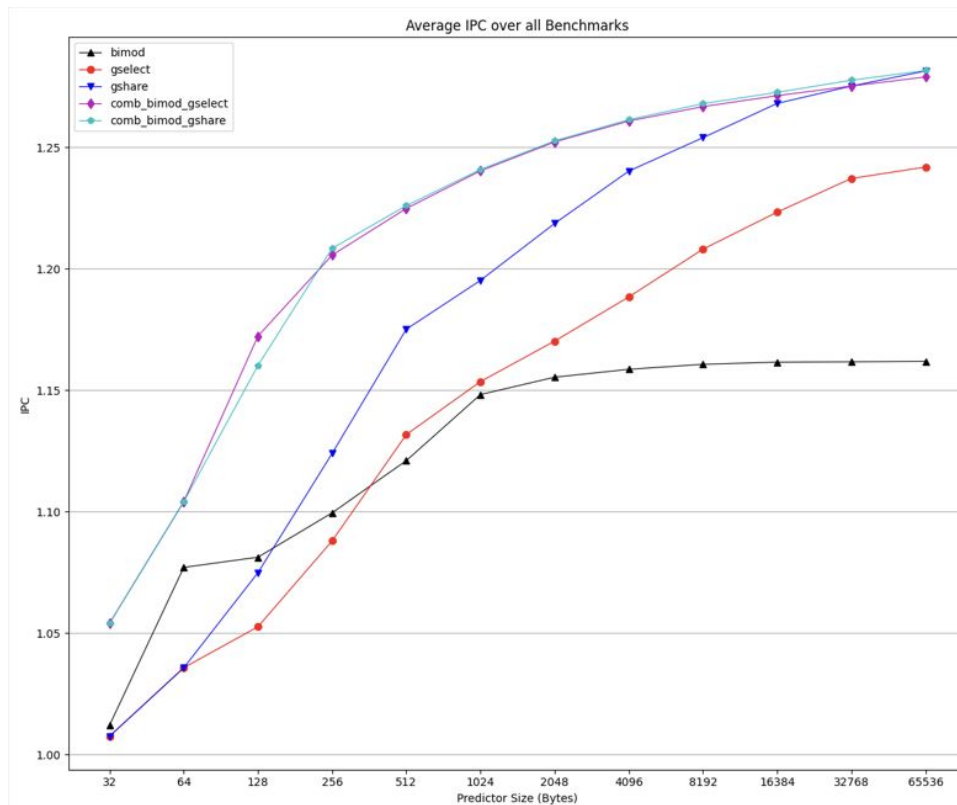




# Results - Accuracy vs Predictor Size



# Results - IPC vs Predictor Size





# References

[1] S. McFarling, "Combining branch predictors," in *Proc. 6th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS VI)*, 1993, pp. 213-223. doi: 10.1145/166321.166343

[2] T. M. Austin, *A User's and Hacker's Guide to the SimpleScalar Architectural Research Tool Set*, Intel MicroComputer Research Labs, Jan. 1997.

**Any ?s**

**Thank You!**

