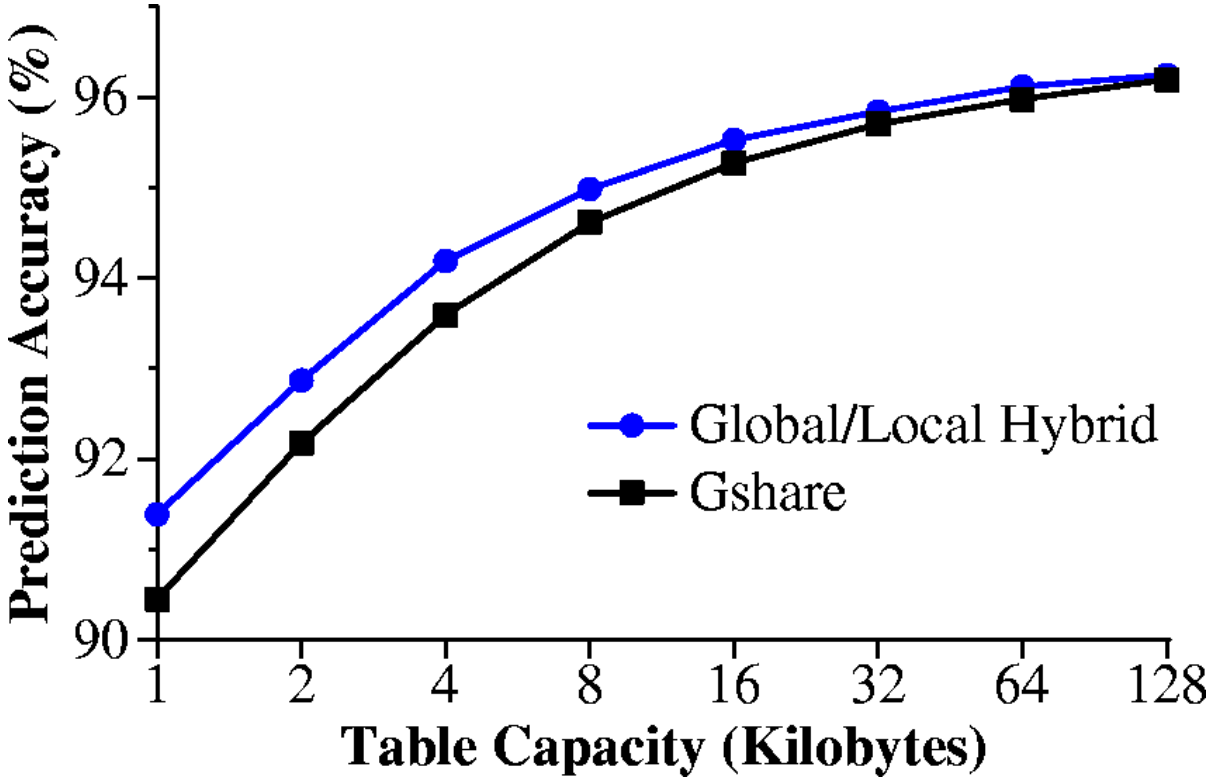
The Impact of Delay on the Design of Branch Predictors

Daniel A. Jiménez
Stephen W. Keckler
Calvin Lin

Department of Computer Sciences
The University of Texas at Austin
Austin, Texas

Motivation

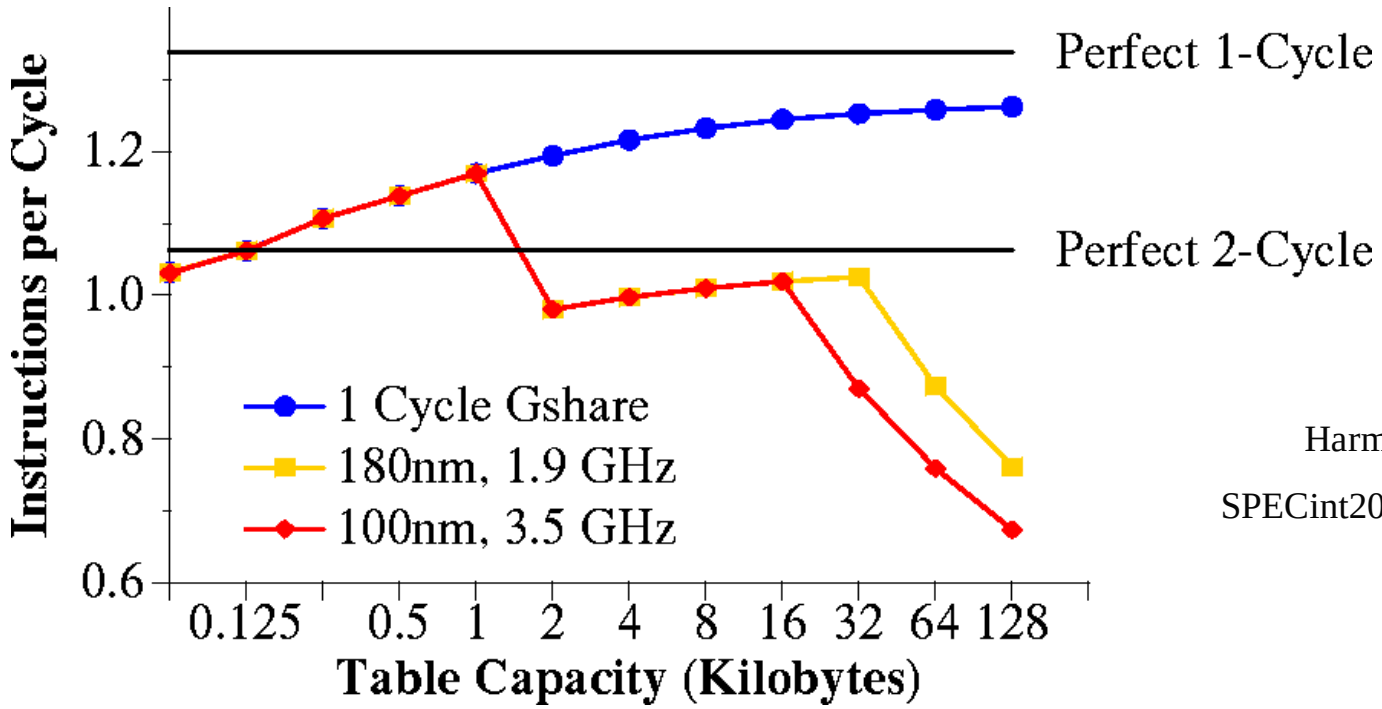
- Branch prediction is important
- Higher clock rates, wide-issue, deeper pipelines
- Accuracy increases with table size



Higher is always better

Motivation

- But delay also increases with table size
- Conclusion: need single-cycle prediction



Harmonic Mean,

SPECint2000, SimpleScalar

The Challenge

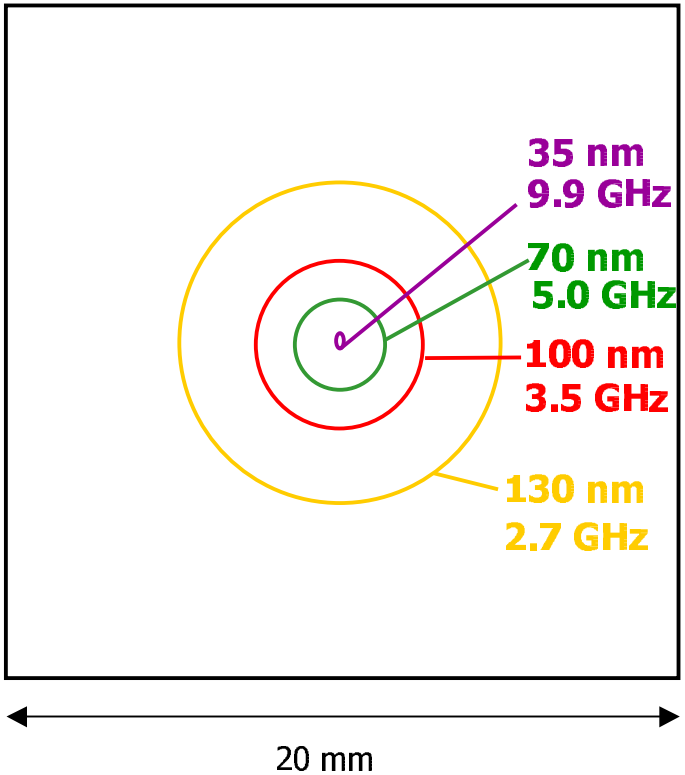
- Competing goals:
  - Accuracy of large structures
  - Single-cycle prediction
  - Solution scalable to future technologies

Overview

- Background
- Mechanisms for tolerating delay
  - Caching
  - Overriding
  - Cascading
- Methodology
- Results and Evaluation

Clock Rate and Technology Scaling

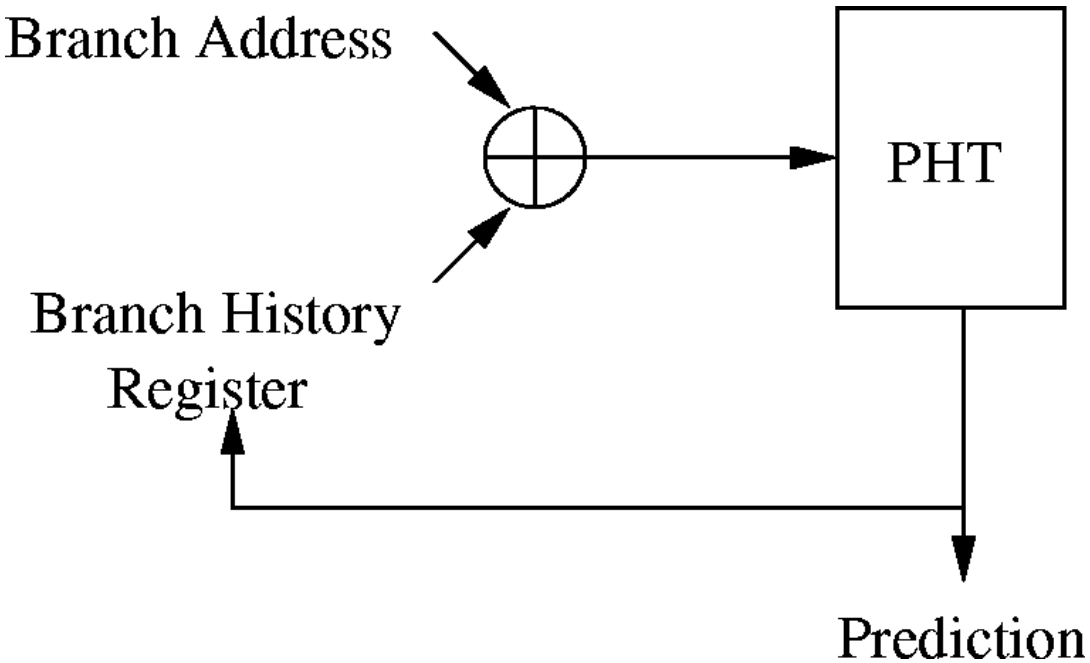
- Aggressive clocking + wire delay
- Less area reachable in one cycle



[Agarwal et al 2000]

gshare: Our Baseline Predictor

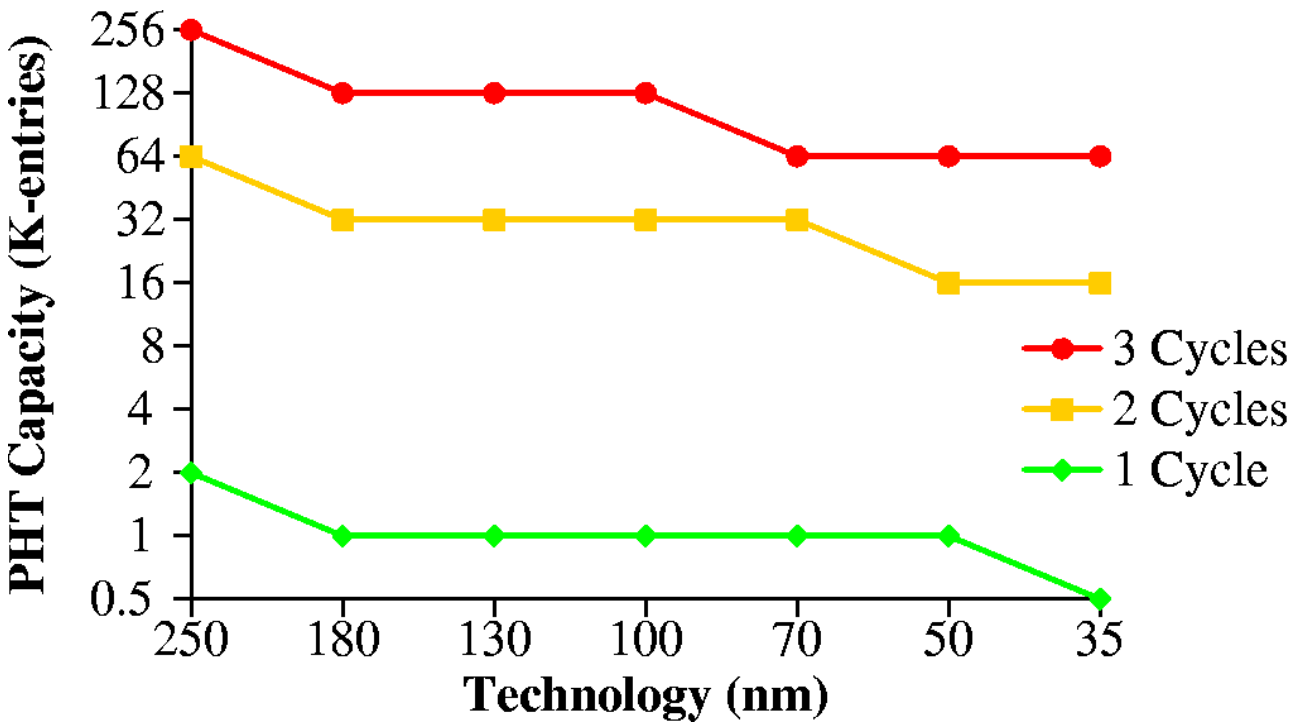
- Pattern history table (PHT) learns correlations
- PHT size is main source of delay



[McFarling 1993]

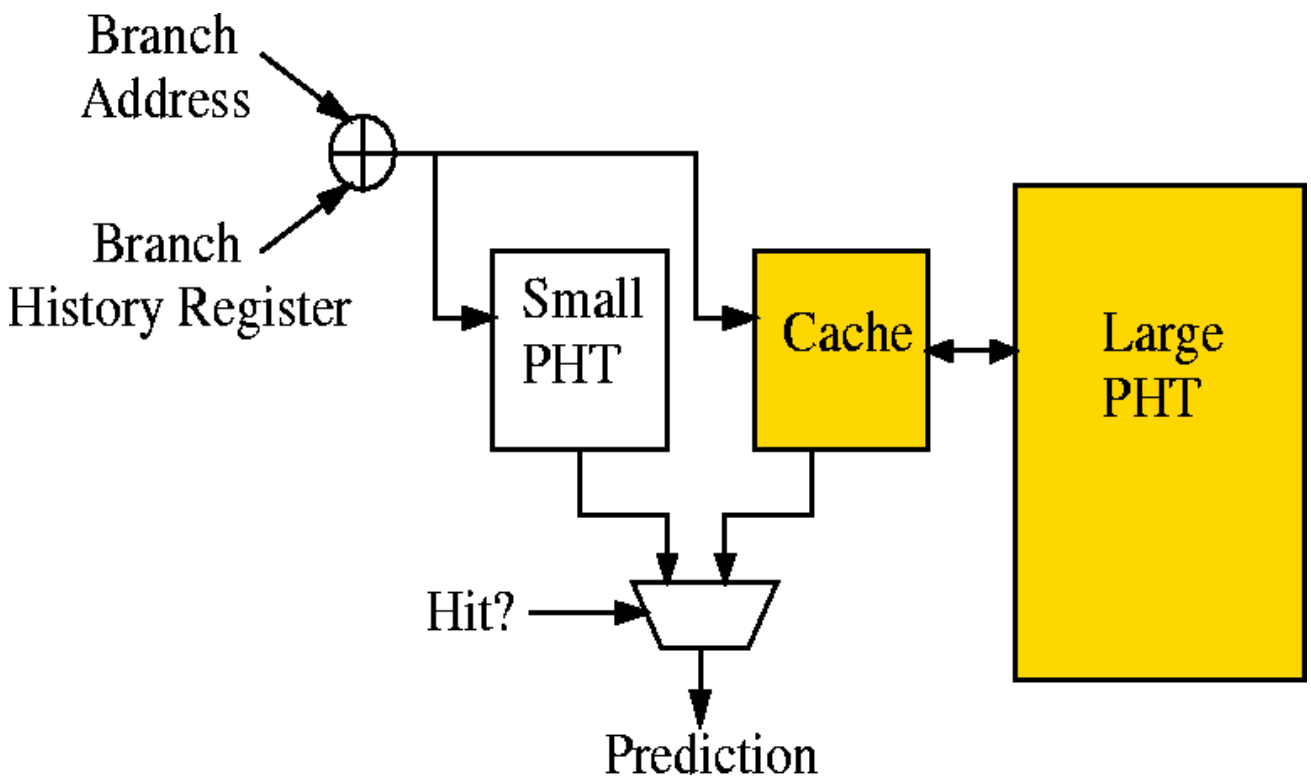
Technology Impact on Branch Prediction

- Aggressive clock rate + wire delay
- PHT size must decrease for single-cycle operation
- How can we use larger, slower structures?



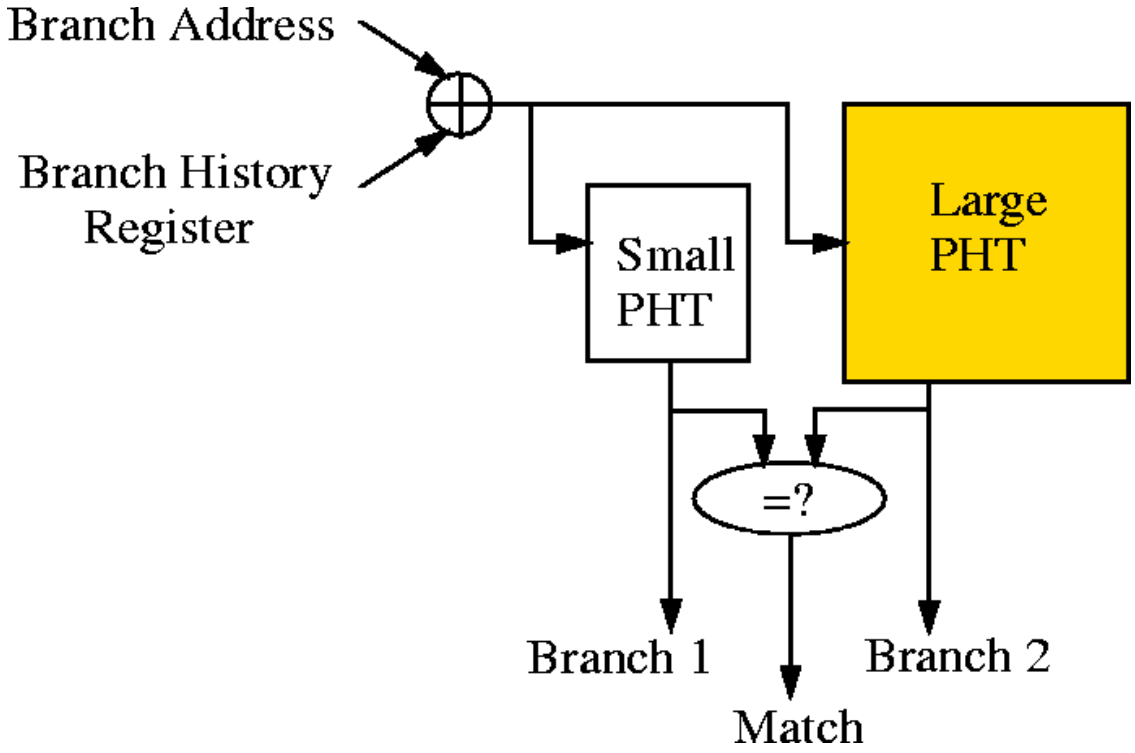
Design I: Caching Predictor

- Entries from a large, slow PHT are cached
- Quicker PHT is used on a cache miss



Design II: Overriding Predictor

- Fast PHT returns a quick initial prediction
- If slow PHT disagrees, override prediction
- Inspired by Alpha 21264

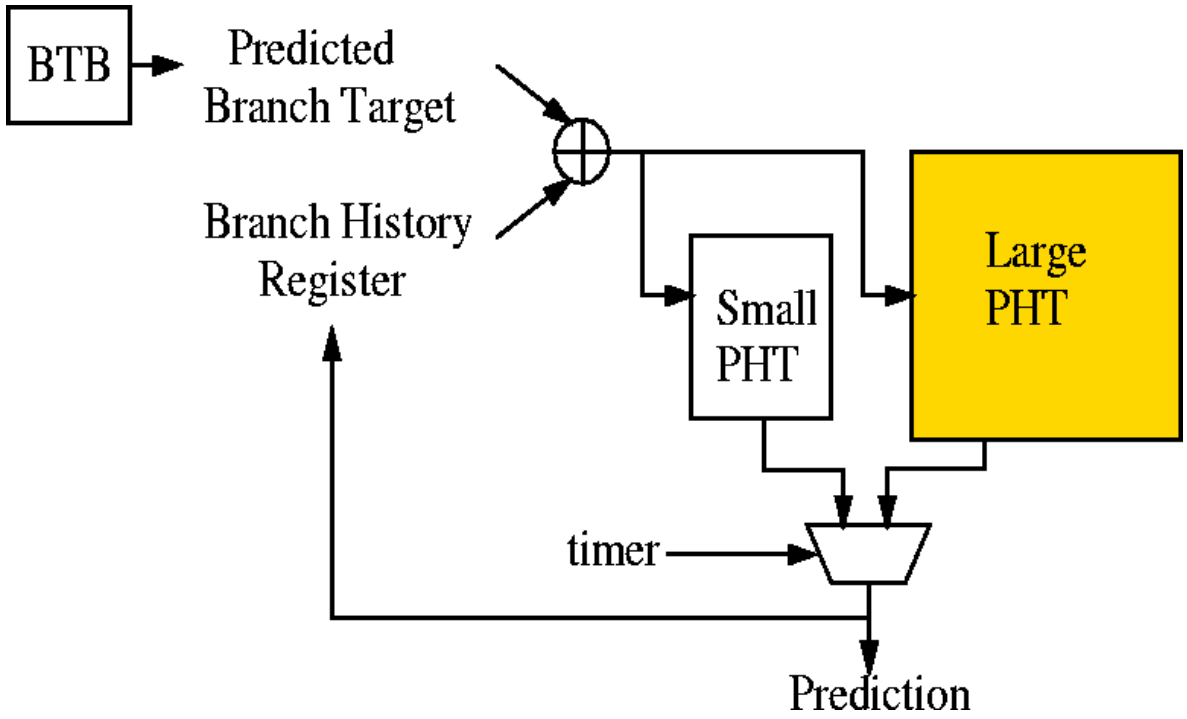


Design III: Cascading Predictor

- Lookahead predictor uses time between branches
- Over 60% of branches come more than one cycle apart
- Uses larger structures until result needed

lookahead:
[Seznec `96]
[Onder `99]

cascading:
[Drisen `98]
[Evers 2000]



Methodology

- Simulation framework:
  - Modified Cacti cache simulator [Wilton & Jouppi 1995]
  - Modified SimpleScalar [Burger & Austin 1997, Agarwal 2000]
  - SPECint2000 with ref inputs
- Parameters tuned for best IPC for each predictor:
  - History length
  - Structure sizes and delays
- Also simulated a 21264-like hybrid

Best Configurations for Overriding

Tech (nm)	Clock Rate (GHz)		#Entries, Fast PHT	#Entries, Slow PHT	Slow PHT Delay (clks)
250	1.4		2K	64K	2
180	1.9		1K	128K	3
130	2.7		1K	32K	2
100	3.5		1K	32K	2
70	5.0		1K	64K	3
50	6.9		1K	64K	3
35	9.9		512	16K	2

Best Configurations for Overriding

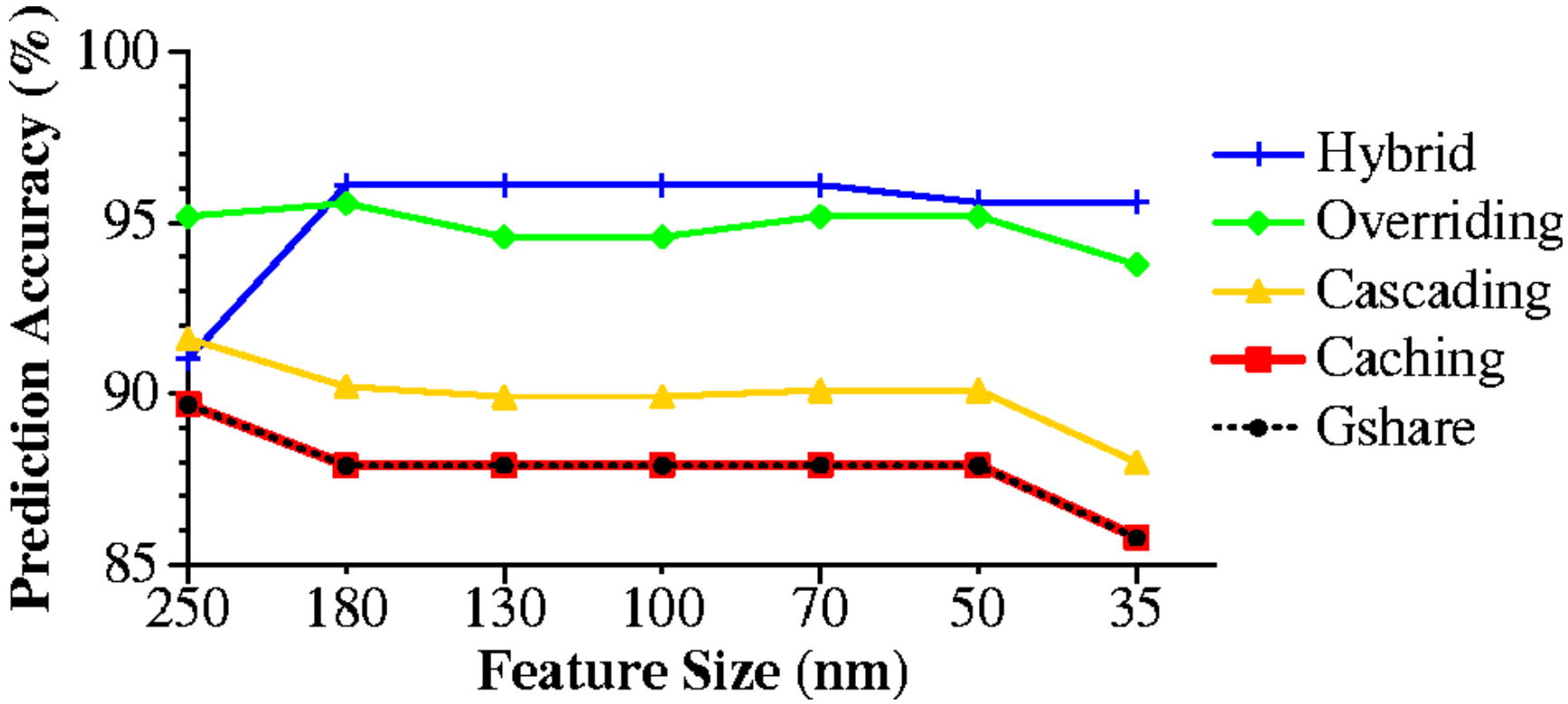
Tech (nm)	Clock Rate (GHz)		#Entries, Fast PHT	#Entries, Slow PHT	Slow PHT Delay (clks)
250	1.4		2K	64K	2
180	1.9		1K	128K	3
130	2.7		1K	32K	2
100	3.5		1K	32K	2
70	5.0		1K	64K	3
50	6.9		1K	64K	3
35	9.9		512	16K	2

Best Configurations for Overriding

Tech (nm)	Clock Rate (GHz)		#Entries, Fast PHT	#Entries, Slow PHT	Slow PHT Delay (clks)
250	1.4		2K	64K	2
180	1.9		1K	128K	3
130	2.7		1K	32K	2
100	3.5		1K	32K	2
70	5.0		1K	64K	3
50	6.9		1K	64K	3
35	9.9		512	16K	2

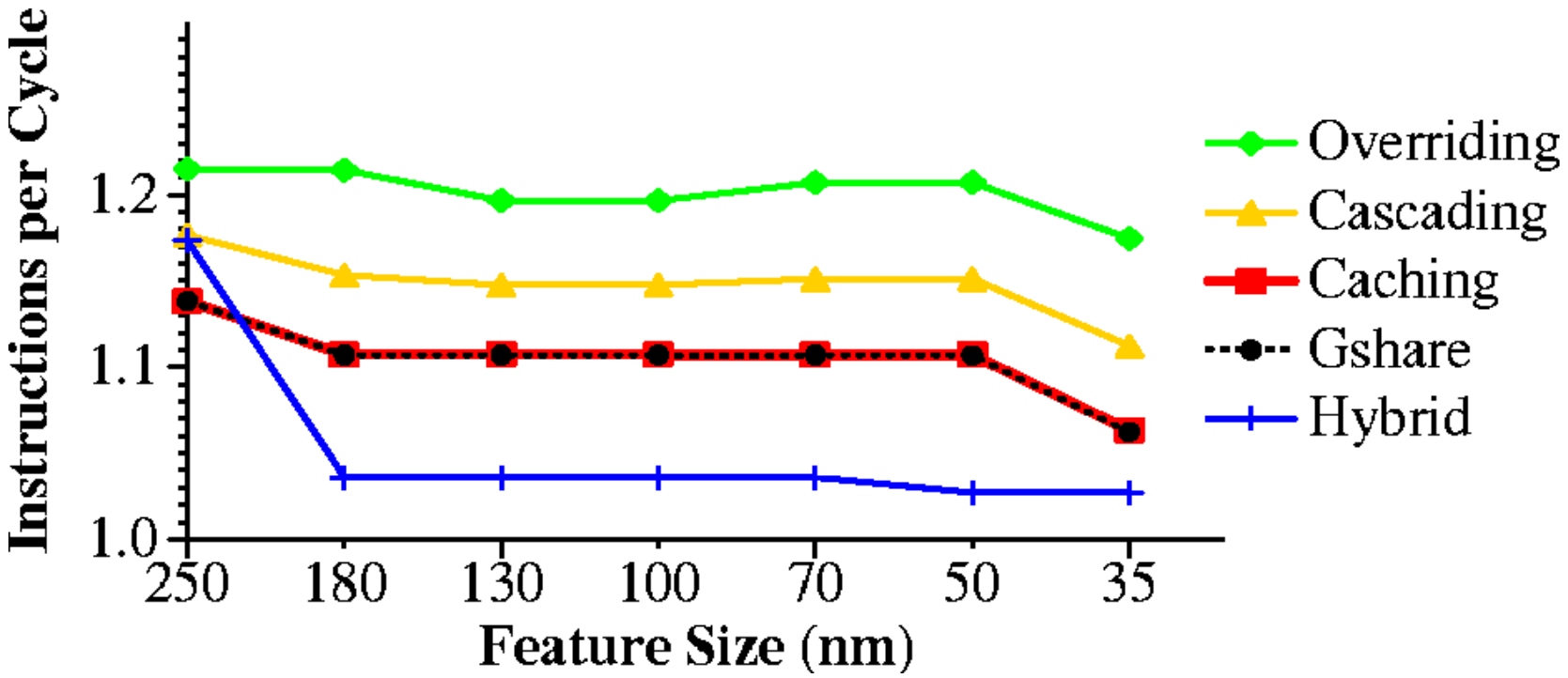
Results: Technology vs. Accuracy

- Traditional measure of predictor performance



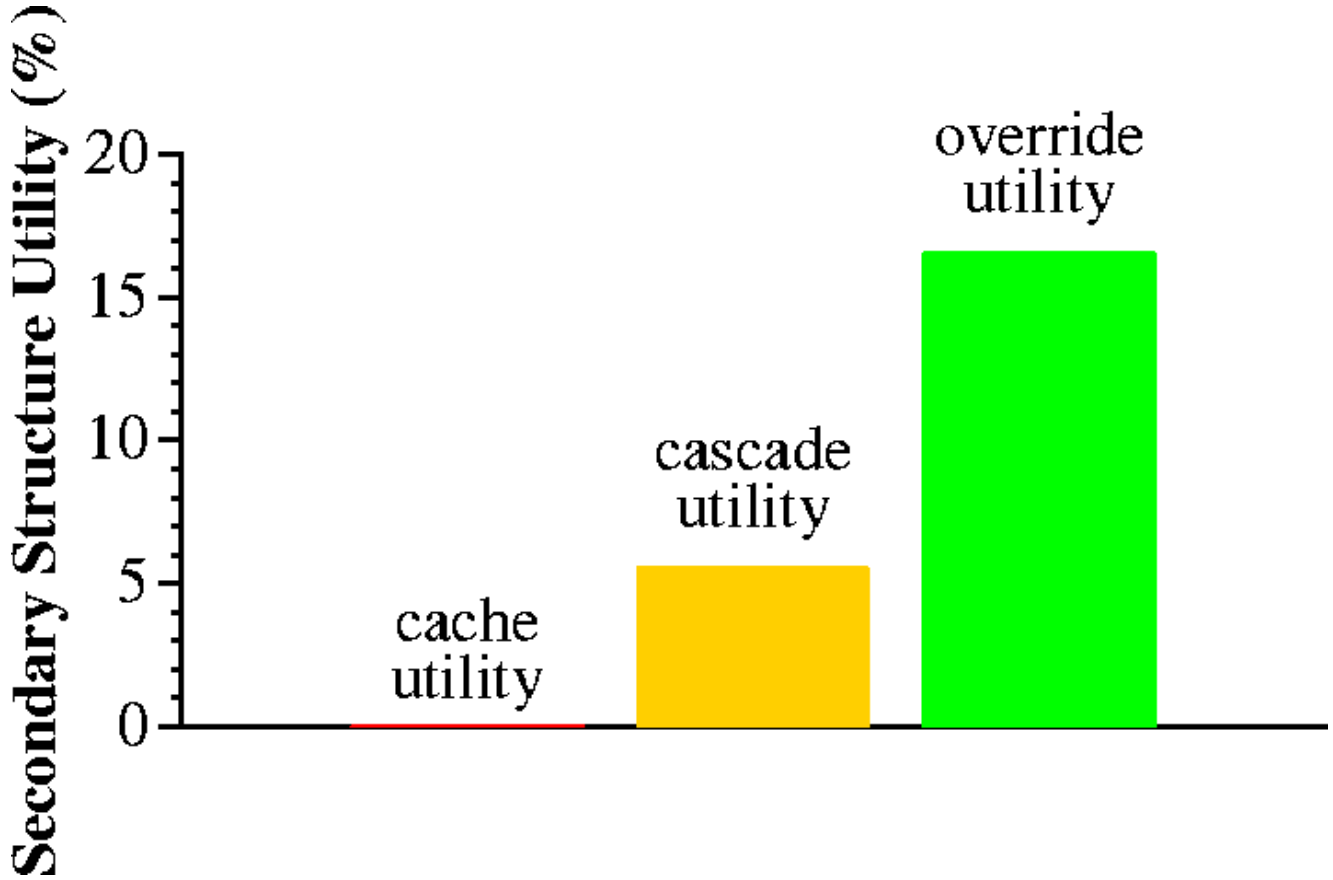
Results: Technology vs. IPC

- Takes delay into account



Results: Structure Utility Rates

- *Utility*: fraction of time second level is useful



Best Configurations for Caching

Tech (nm)	# Tag Bits	# Cache Entries	Cache Entry Width	Large PHT Size
250	7	512	2	64K
180	7	256	2	32K
130	9	256	2	128K
100	10	256	2	256K
70	7	256	2	32K
50	6	256	2	16K
35	7	128	2	16K

Conclusions

- Delay matters!
- Large structures can be used
- Overriding and cascading are feasible

Future Work

- Explore other predictors
- Combine cascading with overriding
- Try using smarter, slower predictors