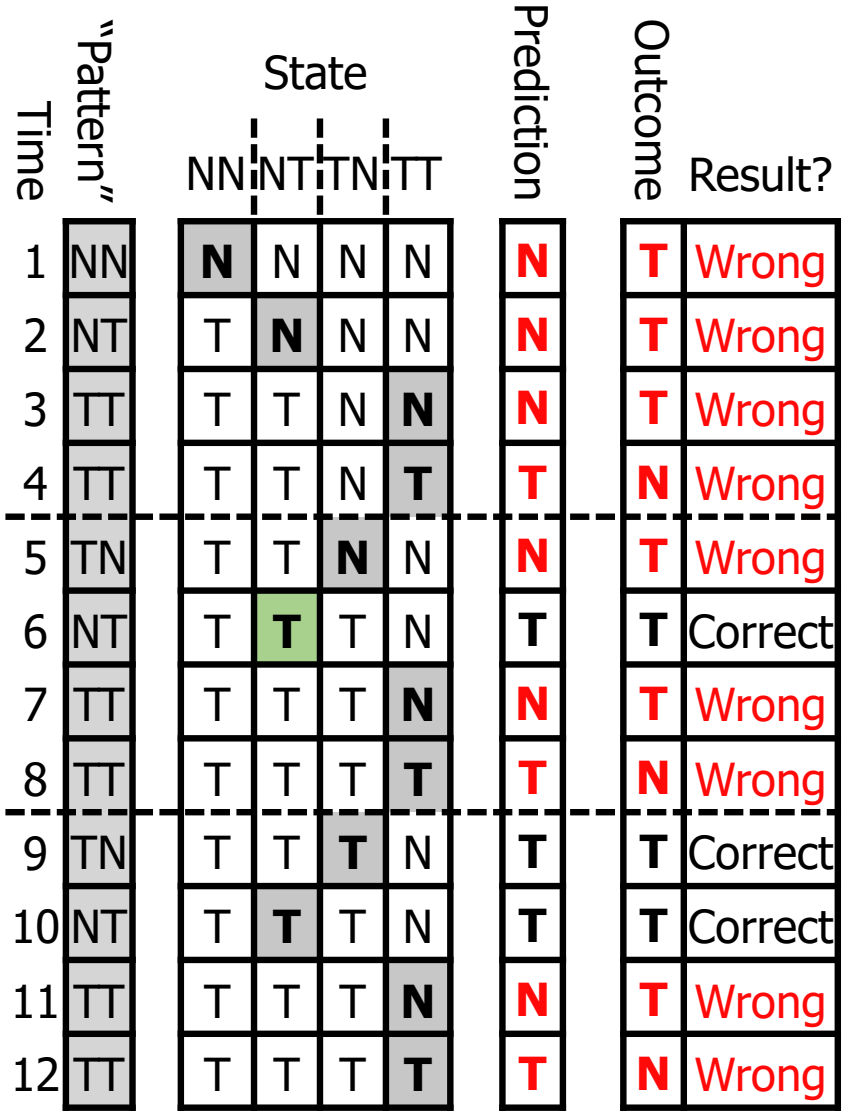
# Correlated predictor

- Prediction with correlation
- *(m,n)* predictor
  - *m* bits of correlation
  - *n*-bit predictor for branch
  - last *m* branches (*2m*) each with an *n*-bit predictor
- • Implementation: Global history with selected address bits (so called "gselect")
  - *m*-bit shig register holds outcome of last *m* branches
  - BHT indexed by *m:low(PC)*
  - BHT can also be indexed just by *m* (global history prediction)

# Correlated Predictor

- **Correlated (two-level) predictor** [Patt 1991]
  - Exploits observation that branch outcomes are correlated
  - Maintains separate prediction per (PC, BHR) pairs
    - **Branch history register (BHR)**: recent branch outcomes
  - Simple working example: assume program has one branch
    - BHT: one 1-bit DIRP entry
    - BHT+**2BHR**: $2^2$ = **four** 1-bit DIRP entries
  - Why didn't we do better?
    - BHT not long enough to capture pattern



| Time | "Pattern" | State NN | State NT | State TN | State TT | Prediction | Outcome | Result? |
|---|---|---|---|---|---|---|---|---|
| 1 | NN | **N** | N | N | N | **N** | **T** | Wrong |
| 2 | NT | T | **N** | N | N | **N** | **T** | Wrong |
| 3 | TT | T | T | N | **N** | **N** | **T** | Wrong |
| 4 | TT | T | T | N | **T** | **T** | **N** | Wrong |
| 5 | TN | T | T | **N** | N | **N** | **T** | Wrong |
| 6 | NT | T | **T** | T | N | **T** | **T** | Correct |
| 7 | TT | T | T | T | **N** | **N** | **T** | Wrong |
| 8 | TT | T | T | T | **T** | **T** | **N** | Wrong |
| 9 | TN | T | T | **T** | N | **T** | **T** | Correct |
| 10 | NT | T | **T** | T | N | **T** | **T** | Correct |
| 11 | TT | T | T | T | **N** | **N** | **T** | Wrong |
| 12 | TT | T | T | T | **T** | **T** | **N** | Wrong |

# Correlated Predictor – 3 Bit Pattern

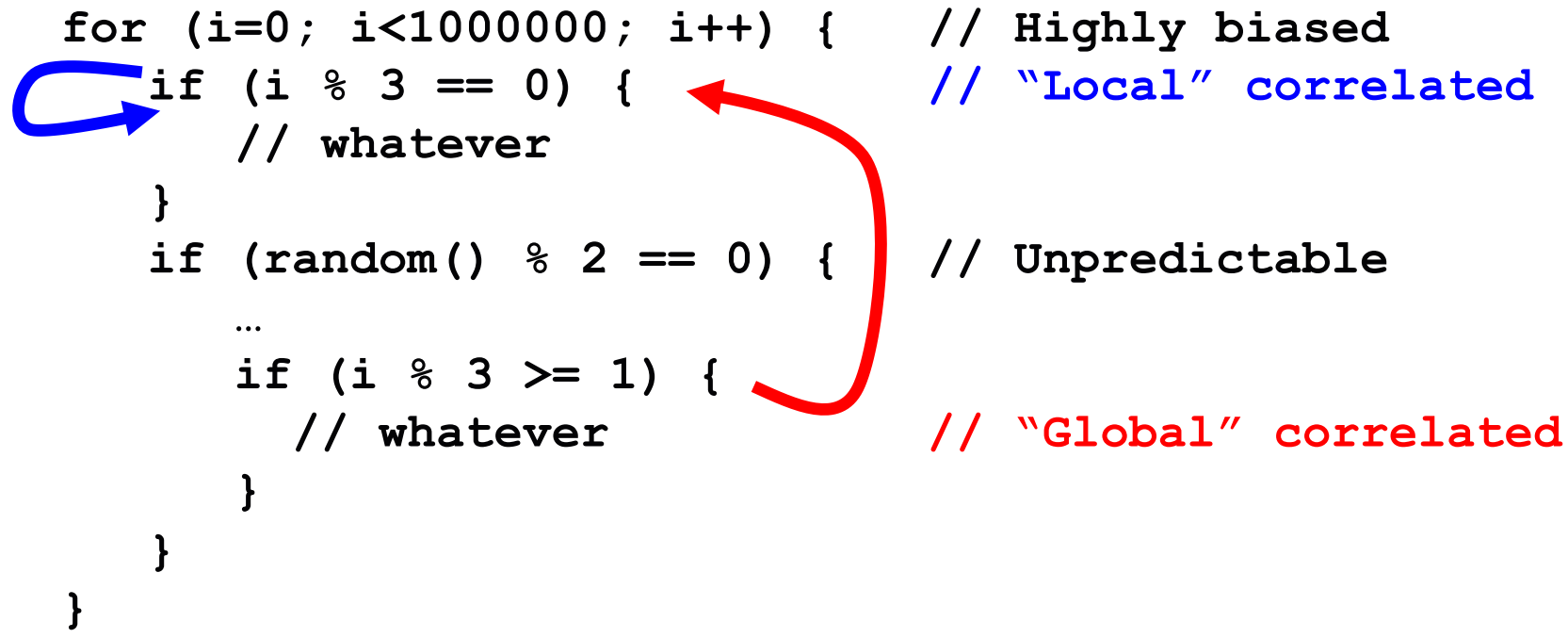
- Try 3 bits of history
- $2^3$ DIRP entries per pattern

| Time | "Pattern" | State: NNN | NNT | NTN | NTT | TNN | TNT | TTN | TTT | Prediction | Outcome | Result? |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | NNN | **N** | N | N | N | N | N | N | N | **N** | **T** | Wrong |
| 2 | NNT | T | **N** | N | N | N | N | N | N | **N** | **T** | Wrong |
| 3 | NTT | T | T | N | **N** | N | N | N | N | **N** | **T** | Wrong |
| 4 | TTT | T | T | N | T | N | N | N | **N** | **N** | **N** | Correct |
| 5 | TTN | T | T | N | T | N | N | **N** | N | **N** | **T** | Wrong |
| 6 | TNT | T | T | N | T | N | **N** | T | N | **N** | **T** | Wrong |
| 7 | NTT | T | T | N | **T** | N | T | T | N | **T** | **T** | Correct |
| 8 | TTT | T | T | N | T | N | T | T | **N** | **N** | **N** | Correct |
| 9 | TTN | T | T | N | T | N | T | **T** | N | **T** | **T** | Correct |
| 10 | TNT | T | T | N | T | N | **T** | T | N | **T** | **T** | Correct |
| 11 | NTT | T | T | N | **T** | N | T | T | N | **T** | **T** | Correct |
| 12 | TTT | T | T | N | T | N | T | T | **N** | **N** | **N** | Correct |

+ No mis-predictions after predictor learns all the relevant patterns!

# Correlated Predictor Design

- Design choice I: one **global** BHR or one per PC (**local**)?
  - Each one captures different kinds of patterns
    - Global history captures relationship among different branches
    - Local history captures "self" correlation
  - Local history requires another table to store the per-PC history
- Consider:



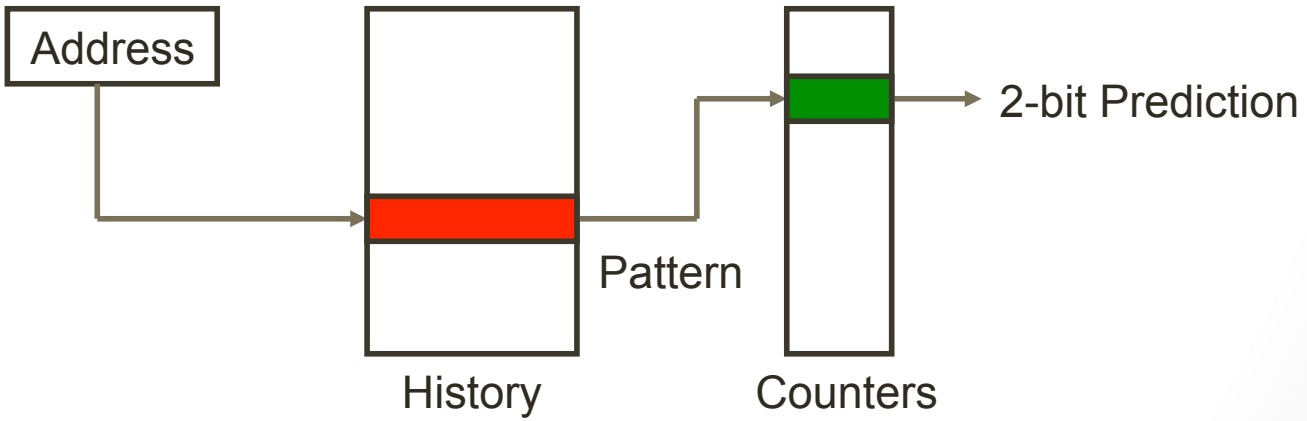
```
for (i=0; i<1000000; i++) {     // Highly biased
    if (i % 3 == 0) {           // "Local" correlated
        // whatever
    }
    if (random() % 2 == 0) {    // Unpredictable
        …
        if (i % 3 >= 1) {
          // whatever           // "Global" correlated
        }
    }
}
```

# Correlated Predictor Design

- Design choice II: how many history bits (BHR size)?
  - Tricky one
  + Given unlimited resources, longer BHRs are better, but…
  – BHT utilization decreases
    – Many history patterns are never seen
    – Many branches are history independent (don't care)
    - PC xor BHR allows multiple PCs to dynamically share BHT
    - BHR length < $\log_2$(BHT size)
  – Predictor takes longer to train
  - Typical length: 8–12

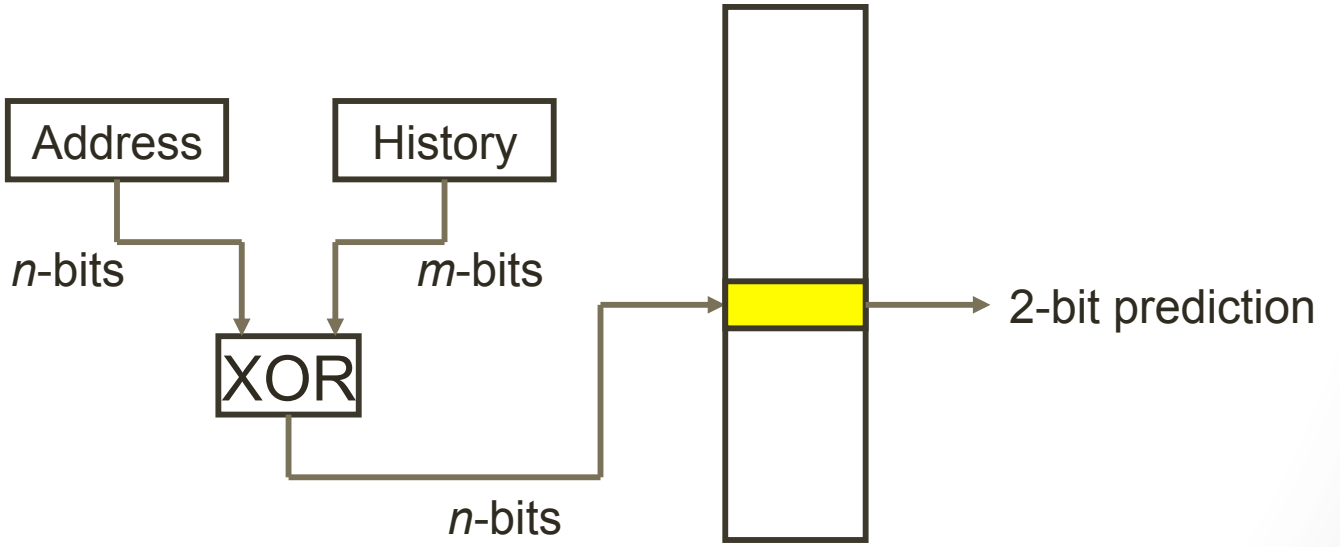
# Local branch prediction

- A two-level history table
- Table 1: history of recent branches indexed by the low address bits of branch instruction PC
- Table 2: two-bit branch predictors indexed by the history from table 1

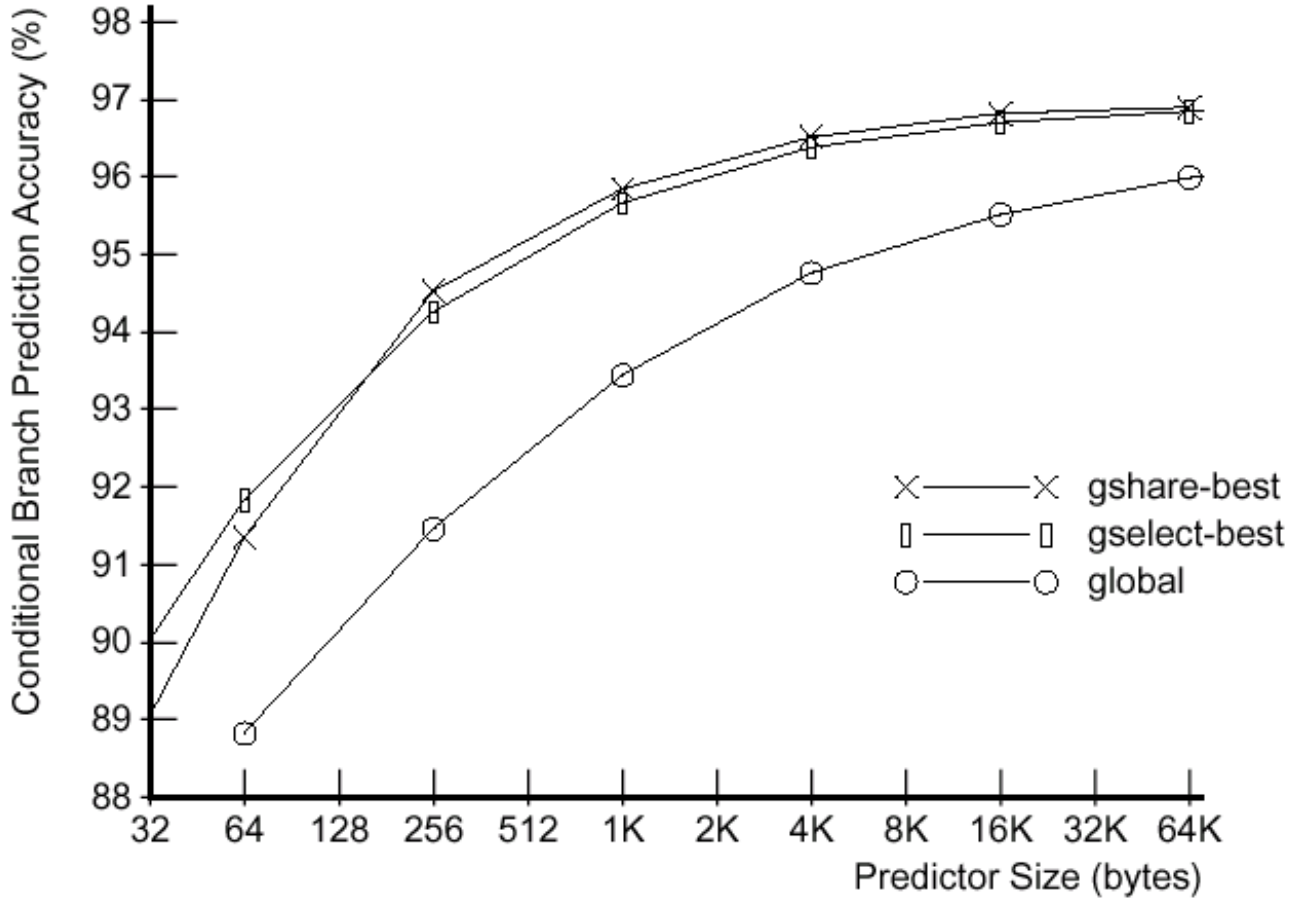


# Gshare – global with index sharing

- Similar to gselect predictor, except the branch address and global history are combined by an XOR

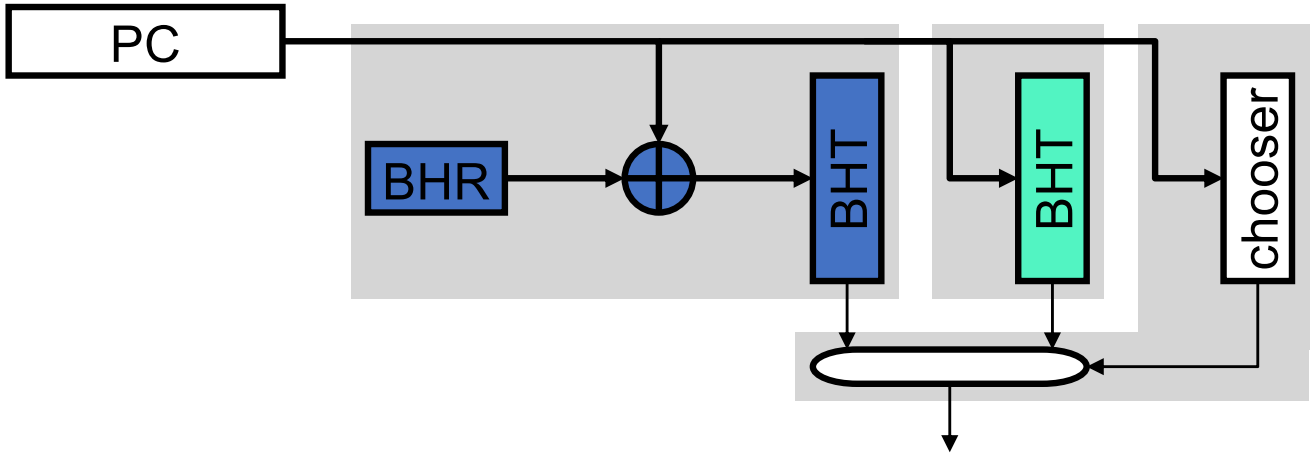


# Gshare vs. Gselect

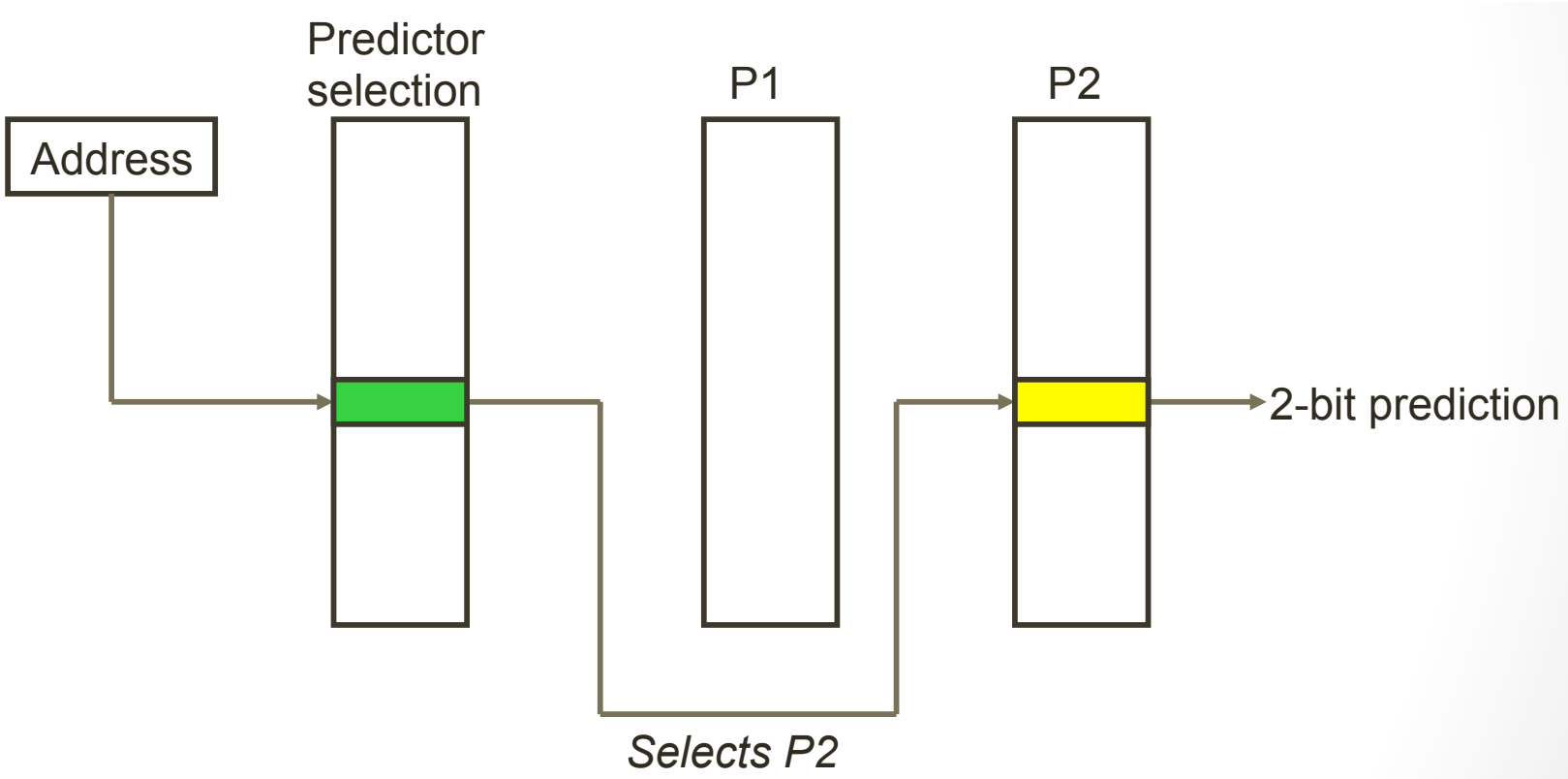


# Tournament branch predictor

- **Hybrid (tournament) predictor** [McFarling 1993]
  - Also known as "combining predictors"
  - Attacks correlated predictor BHT capacity problem
  - Idea: combine two predictors
    - **Simple BHT** predicts history independent branches
    - **Correlated predictor** predicts only branches that need history
    - **Chooser** assigns branches to one predictor or the other
    - Branches start in simple BHT, move mis-prediction threshold
  + Correlated predictor can be made **smaller**, handles fewer branches
  + 90–95% accuracy



# Tournament predictor implementation

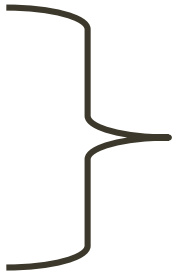


# Keeping track of predictor accuracy

- 2-bit counter incremented/decremented

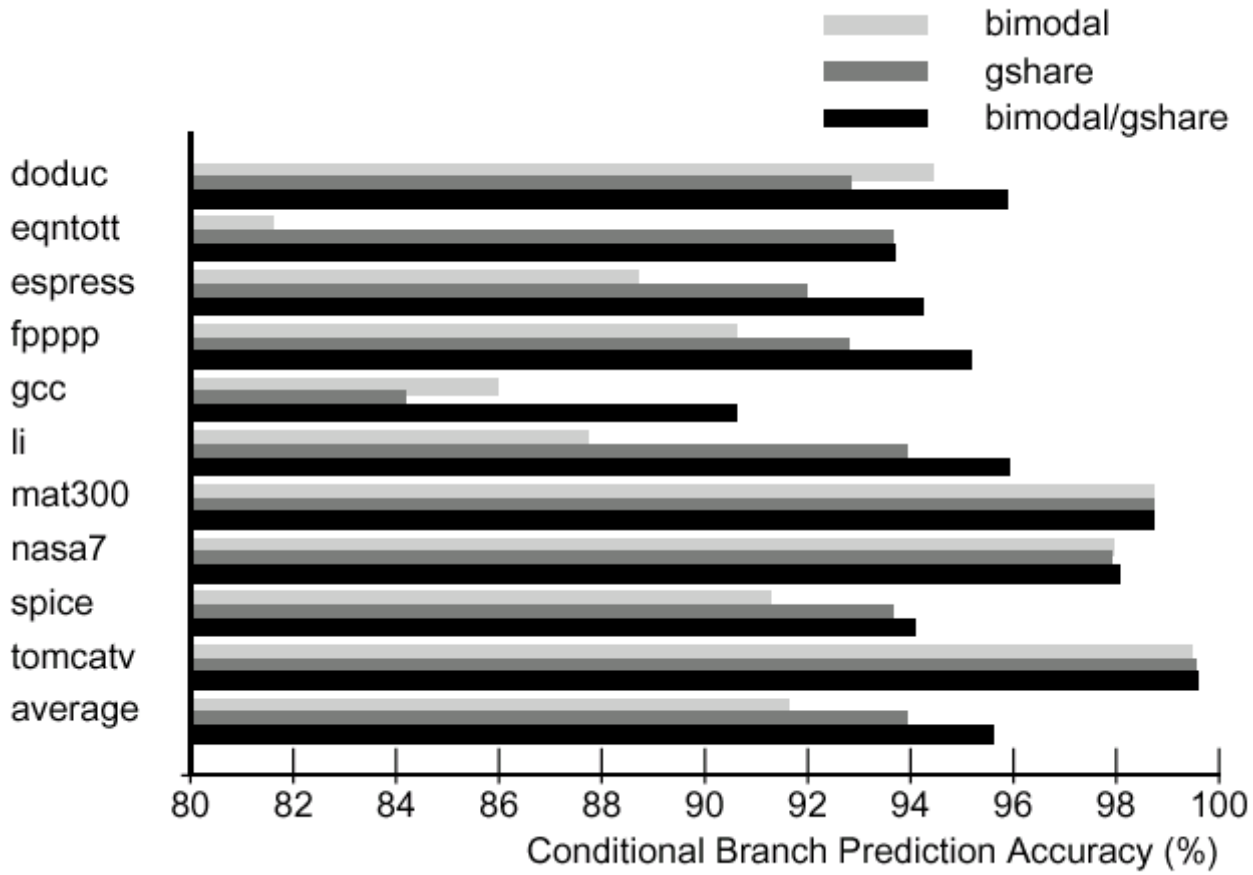
| P1-correct | P2-correct | P1-correct - P2-correct | Action |
|---|---|---|---|
| 0 | 0 | 0-0 = 0 | None |
| 0 | 1 | 0-1 = -1 | Decrement |
| 1 | 0 | 1-0 = 1 | Increment |
| 1 | 1 | 1-1 = 0 | None |

| Counter value | Use predictor |
|---|---|
| 00 | P2 |
| 01 | P2 |
| 10 | P1 |
| 11 | P1 |



Selects which predictor table to use for the prediction

# Tournament predictor performance




Tournament predictor always better than either predictor alone

# Research: Perceptron Predictor

- **Perceptron predictor** [Jimenez]
  - Attacks predictor size problem using machine learning approach
  - History table replaced by table of function coefficients $F_i$ (signed)
  - Predict taken if $\sum(BHR_i * F_i) >$ threshold
  + Table size #PC*|BHR|*|F| (can use long BHR: ~60 bits)
    - Equivalent correlated predictor would be $\#PC * 2^{|BHR|}$
  - How does it learn? Update $F_i$ when branch is taken
    - $BHR_i == 1 \; ? \; F_i{+}{+} : F_i{-}{-}$;
    - "don't care" $F_i$ bits stay near 0, important $F_i$ bits saturate
  + Hybrid BHT/perceptron accuracy: 95–98%