

***Mechanisms: Run-time and  
Compile-time***

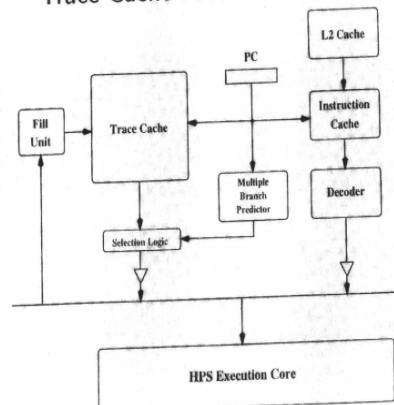
***Mechanisms***

- ***Conditional Branches***
- ***Trace Cache***
- ***Block-Structured ISA***
- ***Wish branches***

## The Conditional Branch Problem

- **Why? Because there are pipelines. (Note: HEP)**
- **Mechanisms to solve it**
  - Delayed branch
  - Take both paths
  - Eliminate branches (predication, compound predicates)
- **Branch Prediction**
  - Static: Always taken, BTFN
  - Early dynamic: LT, 2 bit counter
  - 2 level → gshare, agree, hybrid
  - Indirect jumps
  - Perceptron
  - Expose branch prediction hardware to the software
  - Wish branches

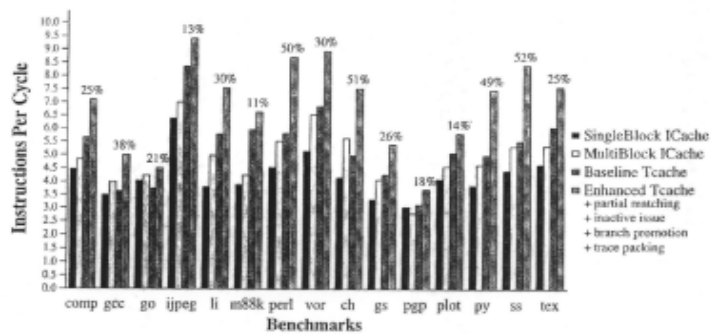
Trace Cache Fetch Mechanism



## Trace Cache Issues

- Storage partitioning
- Set Associativity
- Path Associativity
- Block Collection (retire vs. issue)
- Fill Unit Latency
- Partial Matching
- Inactive Issue
- Dual Path Segments
- Branch Promotion
- Trace Packing

## Overall Performance (aggressive core)



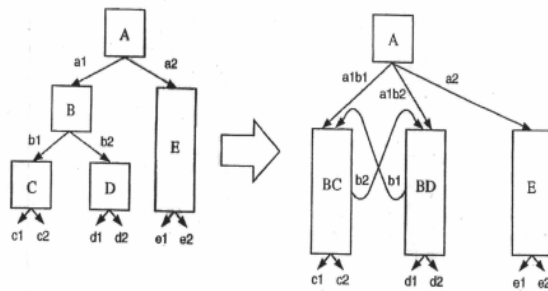
### Block-Structured ISA

- **References:**

- Stephen Melvin (PhD, Berkeley, 1991)
- Eric Hao (PhD, Michigan, 1997)
- ISC paper, Melvin and Patt, 1989
- ISCA paper, Melvin and Patt, 1991
- Micro-29, Hao, Chang, Evers, Patt, 1996

- *The Atomic Unit of Processing*
- *Enlarged Blocks*
- *Savings on Register Pressure*
- *Faulting Branches, Trapping Branches*
- *Serial Execution on Exception*
- *Comparison to Superblocks, Hyperblocks*

### Enlarging Basic Blocks



## **Wish Branches**

- **Compile-time and Run-time**
- **At compile time:**
  - *Does predication even make sense*
  - *If no, regular branch and branch prediction*
  - *If yes, mark wish branch and defer to run-time*
- **At run time:**
  - *Prediction accuracy low: predicate*
  - *Prediction accuracy high: branch predict*