Disjoint Eager Execution: An Optimal Form of Speculative Execution

or: *ILP Speedups in the 10’s*

Augustus (Gus) K. Uht and Vijay Sindagi
Dept. of Elect. and Computer Engr.

UNIVERSITY OF RHODE ISLAND
“A 21st-century microprocessor may well [issue] up to dozens of instructions [per cycle, peak]...”

David A. Patterson, in:
Contributions of the Work

• New form of speculative execution (DEE)
  – Optimal, low cost, high performance:
    
    Speedup factors of 26-31 (2,600% - 3,100%)

• New machine model devised for DEE:
  \texttt{Levo} (target ILP: x 20)
  – On single chip in 4-5 years (by 2000 AD!)
Acknowledgements

• This work supported by the Intel Corp. through a grant from the Intel Research Council. Other support from URI and NSF.

• Other contributors to the work:
  – Sajee Somanathan
  – Sridhar Mahankali
Rest of Talk

• Background
  – ILP limits, Minimal control dependencies, Speculative execution methods

• *Disjoint Eager Execution (DEE)*
  – Theory
  – Heuristic
  – Performance evaluation

• The prototype: Levo
Background

• Oracle ILP speedups:
  – Riseman and Foster (1972), harmonic mean speedup \( S = 25 \);
  – Lam and Wilson (1992): \( S = 159 \); & others....

• w/ realistic constraints, only get: \( S = 2 \) to 3 (to date, using SPECint92’s)

• 50-100 million transistors/chip by 2000 AD

• Instruction set compatibility desirable
Minimal Control Dependencies
(Uht85, Ferrante87, Uht91)

• Classic model: *restrictive control dependencies*

• Can be relaxed: w/MCD, 3 & 4 ind. of 1

1. if (a<8) {
2. \hspace{1em} b=c+d;
}
3. x=y+z;
4. if (p>5) { ... }
Speculative Execution

- Given: $l$ is depth of greatest speculation
- Single Path (SP) - $O(l)$ cost, but low performance: cumulative prob. (cp) $\rightarrow 0$
- Eager Execution (EE) - best performance, w/ infinite resources, but high cost: $O(2^l)$
- Need something better, with good features of both SP and EE:

  *Disjoint Eager Execution (DEE)*
SP and EE Models

Single Path
(l = 6)

Eager Execution
(l = 2)
DEE Theory

• *Branch Path* (resources) definition: dynamic code between branches (PE’s to execute the code in the path as concurrently as possible)

• Rule of Greatest Marginal Benefit:
  
  *Assign resources to most likely paths, over all pending paths*

• Optimal for constrained resources

• Cost: $O(kt^2)$ ; $k<1$
Assigning Resources
Assigning Resources
Assigning Resources
Assigning Resources
Assigning Resources
Assigning Resources
Assigning Resources
Assigning Resources
Comparison of SP, EE and DEE

Eager Execution
(l = 2)

Single Path
(l = 6)

Disjoint Eager Execution
(l = 4)
DEE in Practice

- Problem: hard to compute "true" cumulative probabilities dynamically
- Solution: DEE static tree heuristic:
  - Use average branch prediction accuracy (bpa or $p$) for all branches
  - Static tree shape determined as part of machine design
  - Resources are fixed to the static tree
  - Cost: still $O(kl^2)$ ; $k<1$
**Typical Static Tree**

Total number of branch paths is 34.

A number on a path is the overall probability of the path being executed.
DEE Performance Evaluation

• Method: pixie and modified dsim used
• Assumptions:
  – Unit latency
  – Dynamic Instruction Stream
  – MIPS R3000 instruction set
  – Practical version (heuristic) of DEE modelled
Harmonic Mean Summary

• 5 of 6 SPECint92 benchmarks used:
  • cc1
  • compress
  • eqntott
  • espresso
  • xlisp
  • <=100 million instructions each
• 2-bit saturating counter predictor (Smith81)
• “CD-MF” = “Minimal Control Dependencies”
• “DEE-CD-MF” is DEE with MCD; used in Levo 23
Oracle Speedup: 53.82

Harmonic Mean

Resources (branch paths)

Speedup (factor X sequential)
Comments on Results

• **Speedup factors of 26-31** demonstrated with limited resources and DEE-CD-MF
• Combination of DEE and minimal control dependencies is necessary
• Speedup of 20 potentially achievable with Levo
Levo

- Revised CONDEL-2 (Uht85, Uht92) + DEE
  - From CONDEL-2:
    - IQ: Instruction Queue: static instruction window
    - SSI: register and memory renaming registers
    - ISA: storage addresses, one per SSI
  - Implements: DEE-CD-MF
  - 1-to-1 correspondence with ML and DEE paths of static tree
a. - Broadcast bus for copying of ML state to DEE paths.
b. - Update bus for copying a DEE path state to ML path, upon a DEE branch resolving as mispredicted.

Note: a. and b. can be combined into a single bidirectional bus.
Summary

- **Disjoint Eager Execution (DEE):**
  - Optimal speculative execution
  - Realizes high ILP’s even with hard-to-predict-branch-intensive general-purpose code
  - Achieves 59% of oracle performance
  - Ideas useful elsewhere:
    - Multiprocessors
    - VLIW / software-based ILP machines