Chapter 3 – Instruction-Level Parallelism and its Exploitation (Part 4)

ILP vs. Parallel Computers
Dynamic Scheduling (Section 3.4, 3.5)
Dynamic Branch Prediction (Section 3.3)
Hardware Speculation and Precise Interrupts (Section 3.6)
Multiple Issue (Section 3.7)
Static Techniques (Section 3.2, Appendix H)
Limitations of ILP (Section 3.10)
Multithreading (Section 3.12)
Putting it Together (Mini-projects)

Limits of ILP

How much can ILP buy us?
Limit studies make optimistic assumptions to find the limit for ILP
But may miss impact of compiler, future advances
A highly optimistic study [Wall'93]
Infinite number of physical registers (no register WAW, WAR)
Infinite number of in-flight instructions
Perfect branch prediction
Perfect memory address alias analysis
Single cycle FU
Infinite number of FUs
Single cycle memory (perfect caches)

Limits of ILP (contd.)

(This and next four figures are from an old edition of the book)

Limits of ILP – Impact of Optimistic Assumptions

Limiting Instruction window size
Finding dependences among n instr requires n^2 comparisons
2000 instructions implies 4 million comparisons!
Following use 2K window and 64 issue limit

© 2003 Morgan Kaufmann (USA). All rights reserved.
**Limits of ILP – Impact of Optimistic Assumptions**

Realistic branch prediction
- No charge for mispredictions
- Following use tournament predictor

![Graph showing branch prediction metrics](image1)

Finite registers
- Following uses 256 int and 256 fp for renaming

![Graph showing register usage metrics](image2)

Imperfect memory alias analysis

![Graph showing alias analysis metrics](image3)

**But Limits Studies may be Pessimistic!**

- WAR and WAW hazards through memory
- Unnecessary dependences (e.g., loop iteration count)
- Overcoming data flow limit – value prediction

For more realistic studies
- Address value prediction and speculation
- Speculating on multiple paths
**Multithreading: Instruction + Thread Level Parallelism**

Often superscalar instruction slots are wasted
Why not use them for other threads?

- **Multithreading**
  - Coarse-grained
  - Fine-grained
  - Simultaneous multithreading (SMT) or hyperthreading

(Vs. multiprocessing)

---

**Impact of SMT: 1 vs. 4 threads for TPC-C**

![Impact of SMT](chart)

---

**SMT Speedup & Energy Efficiency: 1 vs. 4 threads**

![SMT Speedup & Energy Efficiency](chart)