Advanced Superscalar Techniques

Outline

- Limits of Instruction-Level Parallelism
- Eager Execution
  References only.
- Load/Store Dependency Prediction and Renaming
  References only.
- Trace Processors
  References only.
- References.

Limits of Instruction-Level Parallelism

From Hennessey and Patterson Section 4.7

Goal: Find issue rate of an ideal processor.

Ideal Processor

- Unlimited number of reservation stations.
- Perfect branch prediction.
- Perfect jump prediction.
- Perfect memory address dependence prediction.

Results:

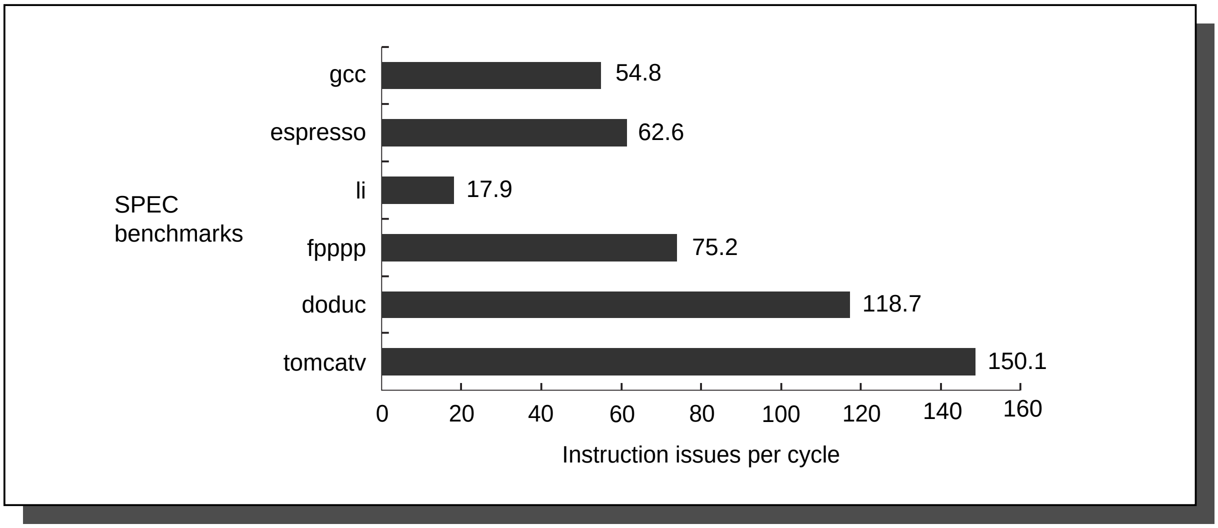

FIGURE 4.38 ILP available in a perfect processor for six of the SPEC benchmarks.

Effect of Window Size

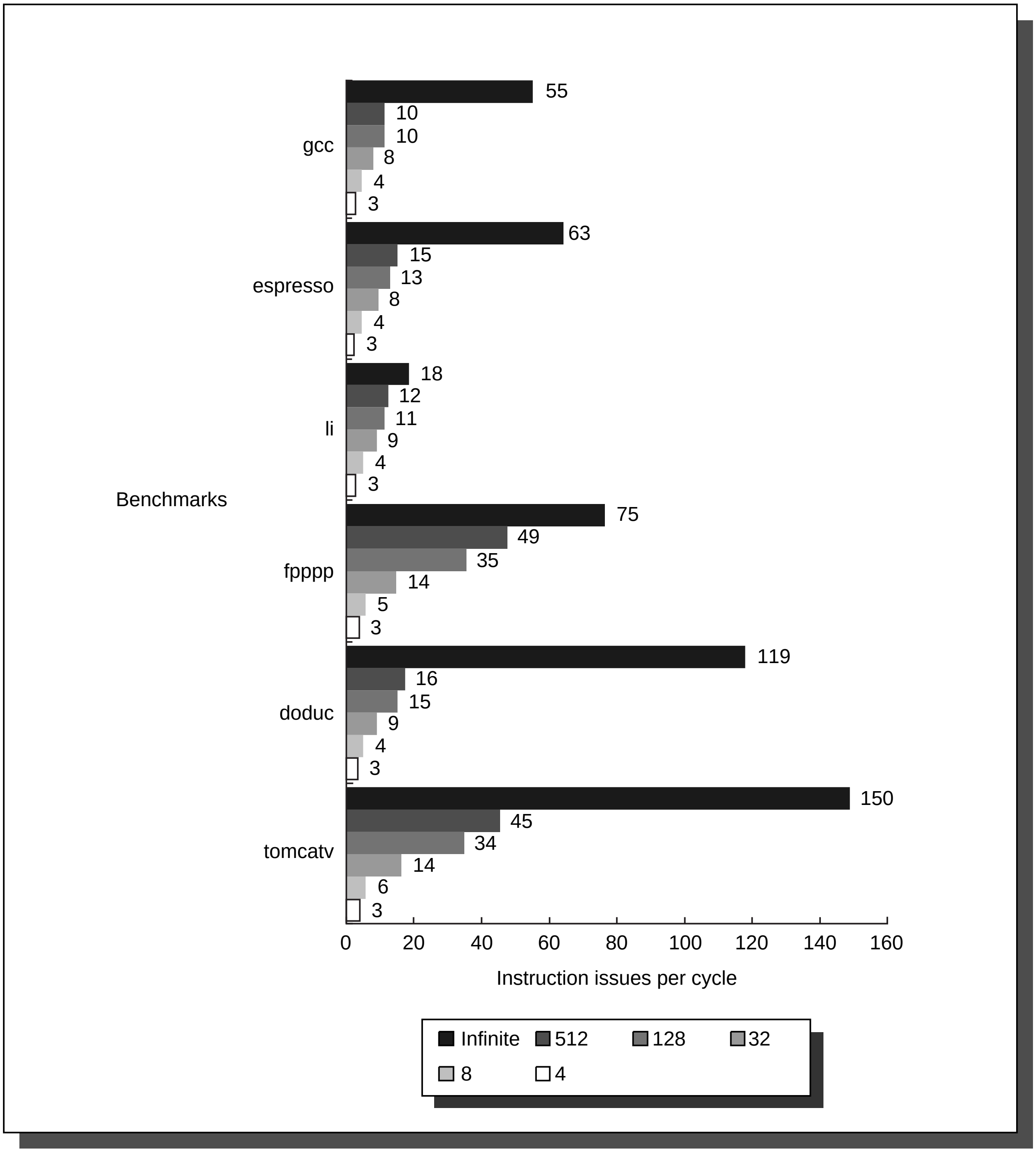

FIGURE 4.40 The effect of window size shown by each application by plotting the average number of instruction issues per clock cycle.

Branch Prediction Effects

Use pretty-good-but-not-ideal processor:

- 2048-instruction window.
- 64-way superscalar.

Branch Predictors

Perfect: all branches predicted.

Selective History: McFarling's gshare/bimodal predictor, 2^{13}-entry tables for gshare, bimodal, and selector.

One-Level: 512-entry BHT.

Static: base predictions on a profile run.

None: No branch prediction.

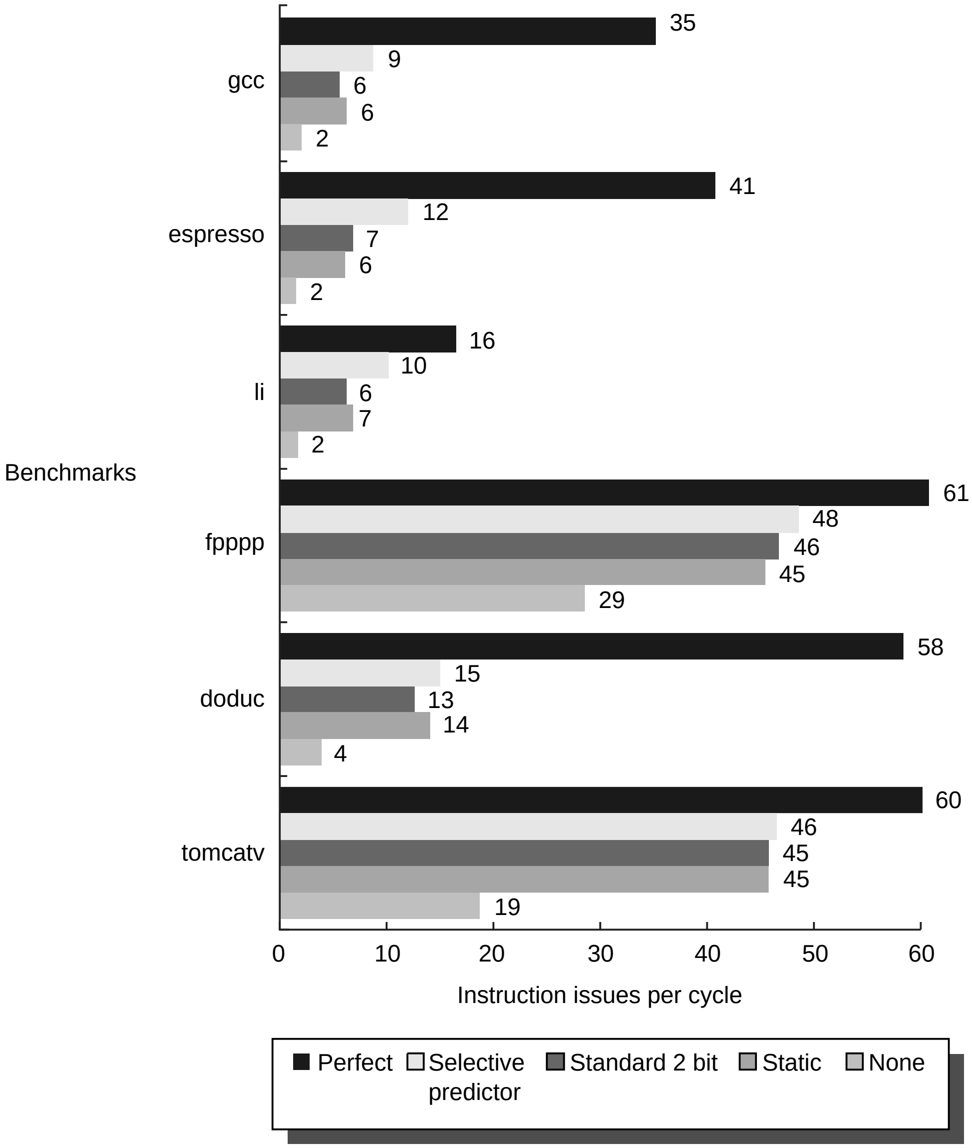

FIGURE 4.42 The effect of branch-prediction schemes sorted by application.

Number of Physical Registers

Vary physical registers. (Effect similar to varying reservation stations.)

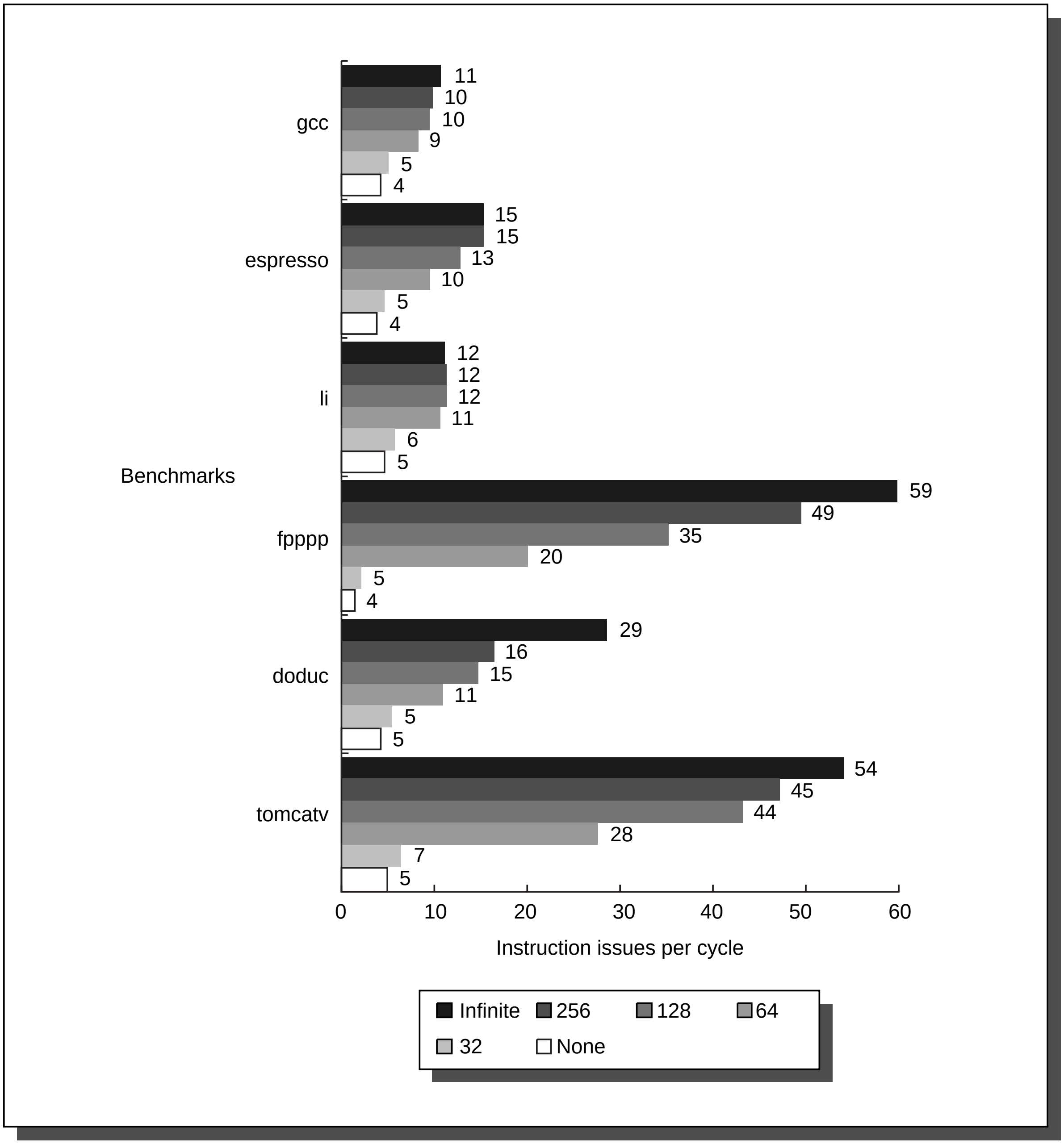

FIGURE 4.44 The reduction in available parallelism is significant when fewer than an unbounded number of renaming registers are available.

Memory Dependency Effects

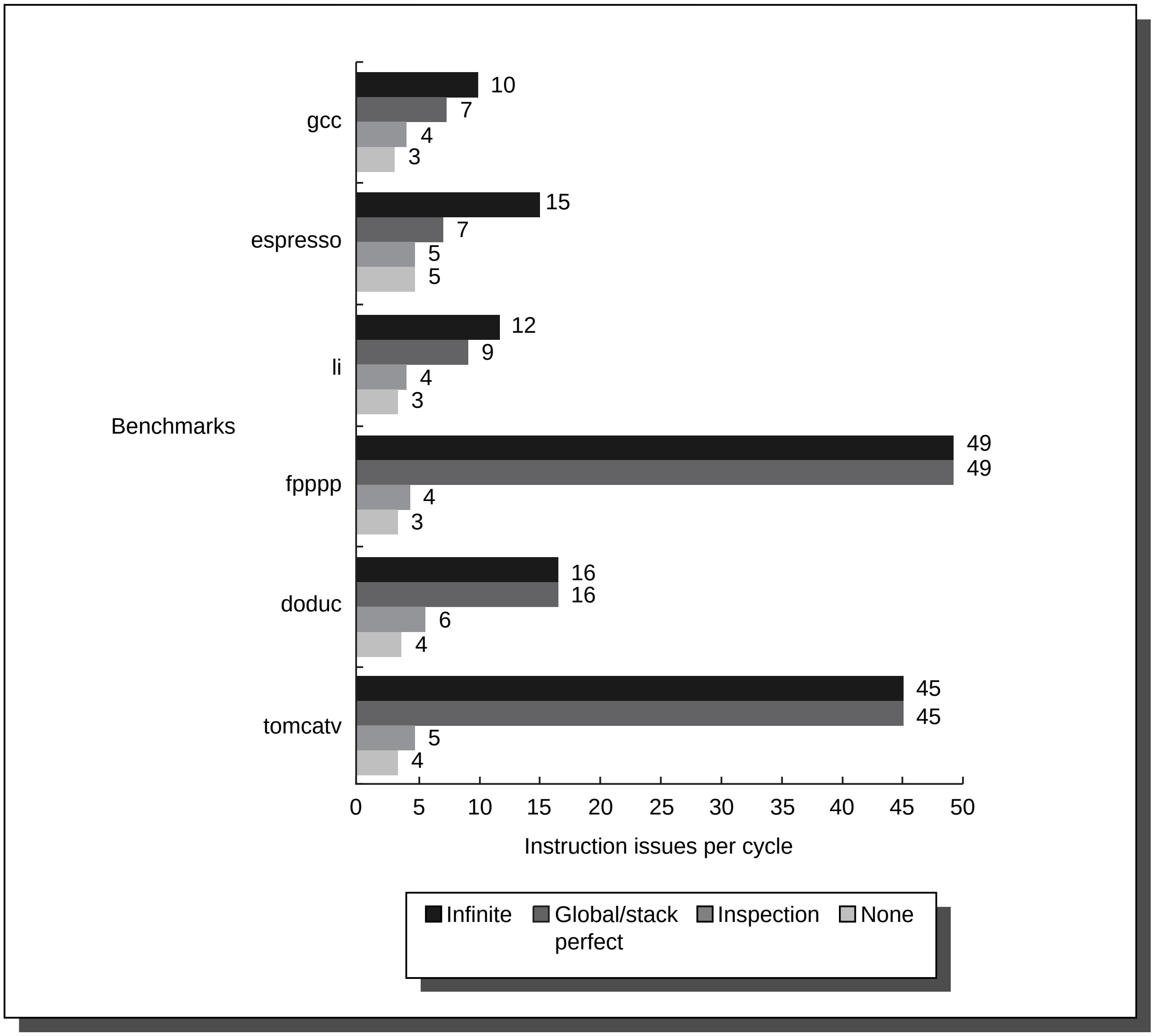

FIGURE 4.46 The effect of varying levels of alias analysis on individual programs.

Window Size for “Realizable” Processors

Realizable Processor

- 64-way superscalar.
- gshare/bimodal predictor with 1024-entry tables.
- Perfect load/store dependency analysis.
- Register renaming with 64 additional registers.

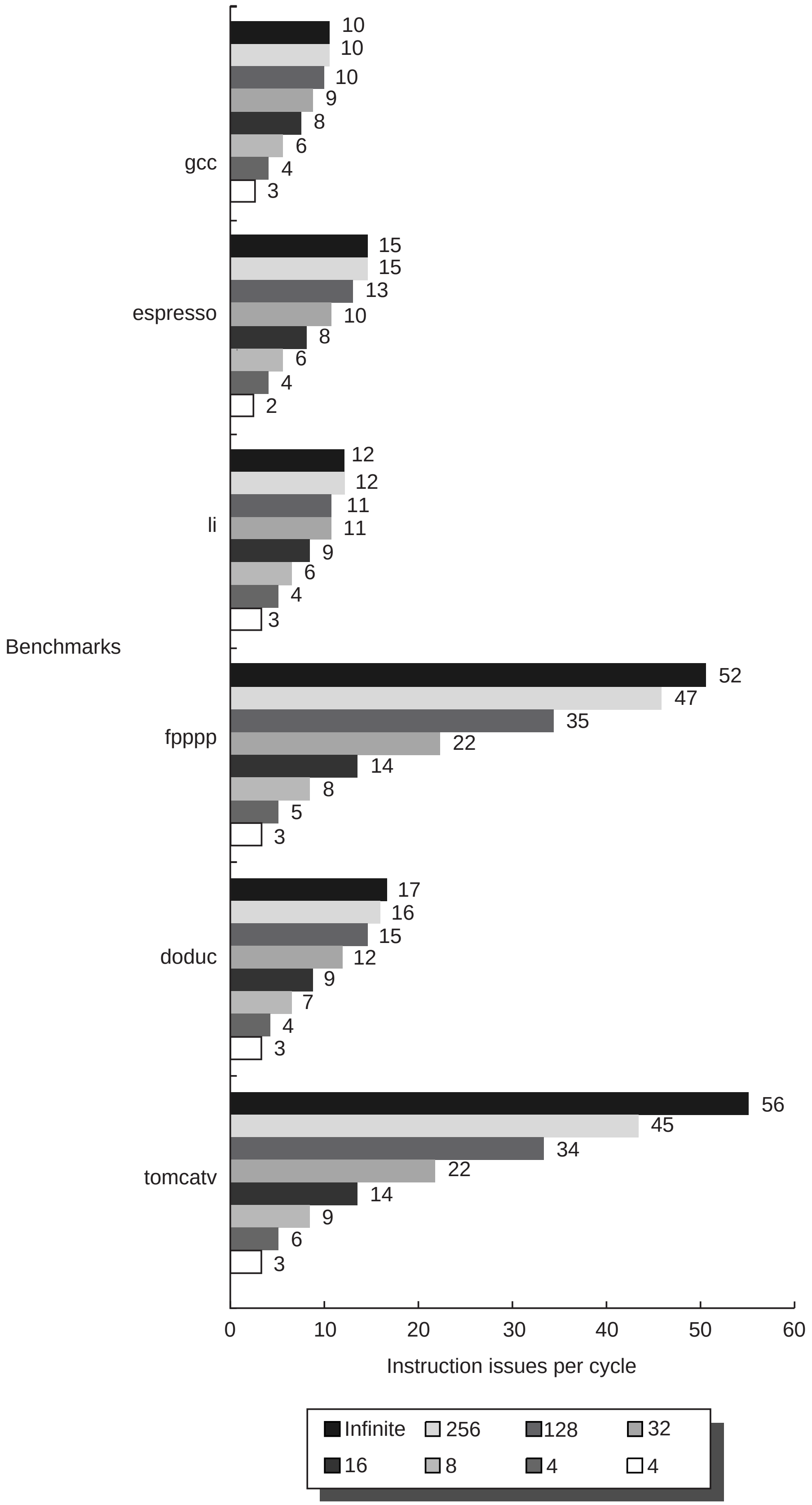

FIGURE 4.48 The amount of parallelism available versus the window size for a

References

Limits of Instruction-Level Parallelism

Material (and graphs) from text, section 4.7:

HP: John L. Hennessy and David A. Patterson, "Computer architecture, a quantitative approach," Palo Alto: Morgan Kaufmann, 1990.

Source used for textbook.

Wall 93: David W. Wall, "Limits of Instruction-Level Parallelism," Technical Report, Digital WRL–93/6, November 1993.

(Selective) Eager Execution

Klauser 98: Artur Klauser, Abhijit Paithankar, and Dirk Grunwald, "Selective eager execution on the PolyPath architecture," in *Proceedings of the International Symposium on Computer Arch.,* June 1998, pp. 250–259.

References, Continued

Load/Store Dependence Prediction and Renaming

Store barrier cache: in load/store queue wait only for stores that had caused dependency violations.

Hesson 97: James H. Hesson, Jay LeBlanc, and Stephen J. Ciavaglia, "Apparatus to dynamically control the out-of-order execution of load-store instructions," US Patent no. 5,615,350, March 1997.

Predicting load/store dependencies using store sets. Includes performance of systems that predict all pairs dependent and no pairs dependent.

Chrysos 98: George Z. Chrysos and Joel S. Emer, "Memory dependency prediction using store sets," in *Proceedings of the International Symposium on Computer Architecture,* June 1998, pp. 142–153.

Forwarding data from store to load if dependency predicted.

Tyson 97: Gary S. Tyson and Todd M. Austin, "Improving the accuracy and performance of memory communication through renaming," Proceedings of the Thirtieth Annual IEEE/ACM International Symposium on Microarchitecture, December 1997, pp. 218–227.

References, Continued

Trace Processors

Description of trace processor using several aggressive techniques, including value prediction. Includes comparison with a higher cost system: a superscalar processor with similar prediction capabilities and issue bandwidth.

Rotenberg 97: Eric Rotenberg, Quinn Jacobson, Yiannakis Sazeides, and Jim Smith, "Trace processors," in *Proceedings of the Proceedings of the Thirtieth Annual IEEE/ACM International Symposium on Microarchitecture,* December 1997, pp. 138–148.

Description of trace processor and comparison to a more limited superscalar processor.

Vajapeyam 97: Sriram Vajapeyam and Tulika Mitra, "Improving superscalar instruction dispatch and issue by exploiting dynamic code sequences," in *Proceedings of the 24th Annual International Symposium on Computer Architecture,* June 1997, pp. 1–12.