A Hardware-Synchronized/Scheduled Multiprocessor Model

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in Electrical Engineering

Nimrod Bayer

Submitted to the Senate of the Technion - Israel Institute of Technology
Shvat 5749 Haifa January 1989

This research was carried out in the Faculty of Electrical Engineering under the supervision of Dr. Ran Ginosar.

My deep gratitude is given to Dr. Ran Ginosar for his dedicated guidance and support through all stages of this research.

The generous financial help of the Gutwirth fund is gratefully acknowledged.

ABSTRACT

This work constitutes the foundation layers of the assessment and development of an architectural model for a parallel computer. The proposed model is a tightly coupled MIMD multiprocessor. A certain amount of affinity exists between the proposed model and Dataflow architectures.

The model's central novel feature is a high flow-rate hardware synchronization/scheduling subsystem. The synchronizer/scheduler implements all the aspects of coordinating interprocessor parallelism: Synchronization, work allocation, and scheduling of tasks according to their urgency. Task allocation is dynamic, and based upon processors availability.

As a run-time working data structure, the synchronizer/scheduler uses the program's task map. The task map is a dependency graph, edited according to defined programming rules. The task map may contain cycles. A particular form of dynamic process generation is supported: Concurrent release of an arbitrary number of processes, all derived from a same task. The dependencies expressed in the task map represent control flow alone, and do not represent data flow. Hence, the proposed architectural model's underlying computation control model is Multi-Threaded Control Flow.

The synchronizer/scheduler comprises a central synchronization/scheduling unit (CSU), and a distribution network which mediates between the CSU and processors. By merging and decomposing synchronization data, the distribution network generates an effect of flow-rate amplification. Synchronization data subject to merge/decomposition concern parcels of processes, which were generated together from the same task.

The hardware and software development aspects of the model are treated within this work at various levels of detail, up to the logic design level.

In addition to the problem of parallel operation coordination, the solutions for several other problems are discussed: Memory latency may be countered, relying on window-based multitasking within processors, and on surplus program parallelism with regard to the number of processors; cache coherence may be maintained

without complex hardware apparatus, relying on shared-variables access rules obeyed by programs; software engineering improvements result from expressing programs using task maps.

The viability of an architectural model's software aspect (its programming model) must be established by benchmark examples. Several benchmark examples, representing various domains, were developed; three of which are included.

The overall manifestation of synchronization/scheduling overhead is an increase in program total execution time. High overhead has strong relation to insufficient synchronization/scheduling flow-rate capability. Lower and upper bounds on the slowdown incurred by a given synchronization flow-rate capability are proved.

Petri networks constitute a common tool for exploring the properties of parallel assemblies. An appendix concerning automatic derivation of Petri networks corresponding to programs coded under the proposed model's conventions is included.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

REFERENCES

מקורות

[1] H. S. Stone, *High-Performance Computer Architecture,* Addison-Wesley, 1987.

[2] M. J. Flynn, "Very High-Speed Computing Systems," *Proceedings of the IEEE* 54, 1966, pp. 1901-1909.

[3] H. T. Kung, "Synchronized and Asynchronous Parallel Algorithms for Multiprocessors," *Algorithms and Complexity*, Academic Press, 1976, pp. 153-200. (Also included in the *Tutorial on Parallel Processing*, edited by R. H. Kuhn and D. A. Padua, Computer Society Press, 1981, pp. 428-463).

[4] N. S. Ostlund, P. G. Hibbard, and R. A. Whiteside, "A Case Study in the Application of a Tightly Coupled Multiprocessor to Scientific Computations," included in *Parallel Computations,* edited by G. Rodrigue, Academic Press, 1982, pp. 315-364.

[5] J. E. Requa and J. R. McGraw, "The Piecewise Data Flow Architecture: Architectural Concepts," *IEEE Trans. on Computers,* Vol. C-32 No. 5, May 1983, pp. 425-438.

[6] Z. Li and W. Abu-Sufah, "A Technique for Reducing Synchronization Overhead in Large Scale Multiprocessors," *Proc. of the 12th Symp. on Computer Architecture,* 1985, pp. 284-291

[7] C. D. Polychronopoulos, D. J. Kuck, and D. A. Padua, "Execution of Parallel Loops on Parallel Processor Systems," *Proc. Int. Conf. on Parallel Processing,* 1986, pp. 519-527.

[8] M. Dubois, C. Scheurich, and F. Briggs, "Synchronization, Coherence, and Event Ordering in Multiprocessors," *IEEE Computer,* Vol. 21 No. 2, February 1988, pp. 9-22.

[9] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAaliffe, and M. Snir, "The NYU Ultracomputer - Designing an MIMD shared Memory Parallel Computer," *IEEE Trans. on Computers,* February 1983, pp. 175-189.

[10] S. P. Midkiff, and D. A. Padua, "Compiler Generated Synchronization for DO Loops," *Proc. Int. Conf. on Parallel Processing,* 1986, pp. 544-551.

[11] P. C. Treleaven, R. P. Hopkins, and P. W. Rautenbach, "Combining Data Flow and Control Flow Computing," *Computer Journal,* Vol. 25 No. 2, 1982, pp. 207-217. (Also included in the *Selected Reprints on Dataflow and Reduction Architectures,* edited by S. S. Thakkar, Computer Society Press, 1987, pp. 355-365).

[12] J. B. Dennis, "The Varieties of Data Flow Computers," *Proc. 1'st Int. Conf. on Distributed Computing Systems,* 1979, pp. 430-439.

[13] K. Hwang, "Multiprocessor Supercomputers for Scientific/Engineering Applications," *IEEE Computer,* Vol. 18 No. 6, June 1985, pp. 57-73.

[14] M. Gransky, I. Koren, and G. M. Silberman, "The Effect of Operation Scheduling on the Performance of a Data Flow Computer," *IEEE Trans. on Computers,* Vol. C-36 No. 9, September 1987, pp. 1019-1029.

[15] J. D. Ullman, "NP-Complete Scheduling Problems," *J. Comput. Syst. Sci.,* Vol 10, June 1975, pp. 384-393.

[16] E. G. Coffman, Ed., *Computer and Job-Shop Scheduling Theory,* New York: Wiley, 1976.

[17] Arvind and R.A. Iannucci, "A Critique of Multiprocessing von Neumann Style," *Proc. 10 Int. Symp. Comp. Arch.,* 1983, pp. 426-436.

[18] M. Katevenis, *Reduced Instruction Set Computer Architectures for VLSI*, MIT Press, 1985.

[19] D. Gelernter, "Domesticating Parallelism," *IEEE Computer*, Vol. 19 No. 8, August 1986, pp. 12-19.

[20] A. H. Karp, "Programming for Parallelism," *IEEE Computer*, Vol. 20 No. 5, May 1987, pp. 43-57.

[21] C. L. Seitz, "System Timing," included as chap. 7 in *Introduction to VLSI Systems* by C. Mead and L. Conway, Addison-Wesley, 1980, pp. 218-254.

[22] S. E. Fahlman, and G. E. Hinton, "Connectionist Architectures for Artificial Intelligence," *IEEE Computer*, Vol. 20 No. 19, January 1987, pp. 100-109.

[23] E. P. Farrel, N. Ghani and P. C. Treleaven, "A Concurrent Computer Architecture and a Ring Based Implementation," *Proc. of the 6'th symp. on Computer Architecture*, 1979, pp. 1-11.

[24] J. D. Brock, A. R. Omondi and D. A. Plaisted, "A Multiprocessor Architecture for Medium-Grain Parallelism," *Proc. 6'th Int. Conf. on Distributed Computing Systems*, 1986, pp. 167-174.

[25] R. Buehrer and K. Ekanadham, "Incorporating Data Flow Ideas into von Neumann Processors for Parallel Execution," *IEEE Trans. on Computers*, Vol. C-36 No. 12 December 1987, pp. 1515-1522.

[26] D. Kuck, *The Structure of Computers and Computations*, Wiley 1978.

[27] C. D. Polychronopoulos and D. J. Kuck, "Guided Self-Scheduling: A Practical Scheduling Scheme for Parallel Supercomputers," *IEEE Trans. on Computers*, Vol. C-36 No. 12 December 1987, pp. 1425-1439.

[28] D. Gajski, D. Kuck, D. Lauwrie, and A. Sameh, "CEDAR," *Report No. UIUCDCS-R-83-1123*, Department of Computer Science, University of Illinois, Urbana, February 1983, pp. 1-25. (Also included in the *Tutorial on Supercomputers: Design and Applications*, edited by K. Hwang, Computer Society Press, 1984, pp. 251-275).

[29] C. Q. Zhu, and P. C. Yew, "A Synchronization Scheme and its Applications for Large Multiprocessor Systems," *Proc. 4th Int. Conf. on Distributed Computing Systems*, 1984, pp. 486-493.

[30] P. Tang and P. C. Yew, "Processor Self-Scheduling for Multiple-Nested Parallel Loops," *Proc. Int. Conf. on Parallel Processing*, 1986, pp. 528-535.

[31] G. F. Pfister et al., "The IBM RP3 Introduction and Architecture," *Proc. Int. Conf. on Parallel Processing*, August 1985, pp. 764-771.

[32] B. J. Smith, "Architecture and Applications of the HEP Multiprocessor Computer System," *Real Time Signal Processing IV, Proceedings of SPIE* August 1981, pp. 241-248. (Also included in the *Tutorial on Supercomputers: Design and Applications*, edited by K. Hwang, Computer Society Press, 1984, pp. 231-238).

[33] B. J. Smith, "A Pipelined, Shared Resource MIMD Computer," *Proc. Int. Conf. on Parallel Processing*, 1978, pp. 6-8. (Also included in the *Tutorial on Advanced Computer Architecture*, edited by D. P. Agrawal, Computer Society Press, 1986, pp. 39-41.

[34] W. Crowther, J. Goodhue, R. Gurwitz, R.Rettberg, and R. Thomas, "The Butterfly Parallel Processor," *Newsletter of the Computer Architecture Technical Comittee* (IEEE Computer Society), September/December 1985, pp. 18-45.

[35] J. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.

[36] T. Etzion and M. Yoeli, "Super-Nets and Their Hierarchy," *Theor. Comp. Sci.* 23, 1983, pp. 243-272.

[37] G. F. Pfister and V. A. Norton, "'Hot Spot' Contention and Combining in Multistage Interconnection Networks," *Proc. 1985 Int. Conf. on Parallel Processing*, Aug. 1985, pp. 790-797.

[38] R. Lee, "On 'hot spot' contention," *ACM SIGARCH Computer Architecture News*, Vol. 13, No. 5, pp. 15-20, Dec. 1985.

[39] R. J. Swan, S. H. Fuller and D. P. Siewiorek, "Cm* — A modular multi-microprocessor," *AFIPS Conf. Proc., 1977 National Computer Conference*, pp. 637-644.