Bibliography

[1] Abu-Sufah, D. Kuck, and D. Lawrie. Automatic program transformation for virtual memory computers. In *Proceedings of the 1979 National 1st Computer Conference*, June 1979.

[2] W. B. Ackerman. Data flow languages. *IEEE Computer*, 15(2):15–25, February 1982.

[3] W. B. Ackerman. *Efficient Implementation of Applicative Languages.* Technical Report 323, Laboratory for Computer Science, MIT, Cambridge, MA, March 1984.

[4] W. B. Ackerman. *A Structure Memory for Data Flow Computers.* Technical Report 186, Laboratory for Computer Science, MIT, August 1977.

[5] W. B. Ackerman. A structure processing facility for data flow computers. In *Proceedings of the 1978 International Conference on Parallel Processing*, August 1978.

[6] W. B. Ackerman and J. B. Dennis. *VAL—A Value-Oriented Algorithmic Language.* Technical Report 218, Laboratory for Computer Science, MIT, 1979.

[7] D. A. Adams. *A Computation Model with Data Flow Sequencing.* Technical Report TR CS-117, School of Humanities and Science, Stanford University, Stanford, CA, December 1968.

[8] G. B. Adams, R. L. Brown, and P. J. Denning. *Report on an Evaluation Study of Data Flow Computation.* Technical Report No. 85.2, Research Institute of Advanced Computer Science, NASA Ames Research Center, April 1985.

[9] T. Agerwala and Arvind. Special issue on data flow systems. *IEEE Computer*, 15(2), February 1982.

[10] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley Publishing Co., 1974.

[11] Arvind and D. E. Culler. Dataflow architectures. *Annual Reviews in Computer Science*, 1:225–253, 1986.

[12] Arvind and D. E. Culler. Managing resources in a parallel machine. In J. V. Woods, editor, *Fifth Generation Computer Architecture*, pages 103–121, Elsevier Science Publishers, 1986.

[13] Arvind et al. *The Tagged Token Dataflow Architecture (preliminary version).* Technical Report, Laboratory for Computer Science, MIT, Cambridge, MA., August 1983.

[14] Arvind, K. P. Gostelow, and W. Plouffe. *An Asynchronous Programming Language and Computing Machine.* Department of Information and Computer Science, University of California, Irvine, 1978.

[15] Arvind and R. A. Iannucci. A critique of multiprocessing von Neumann style. In *Proceedings of the Tenth Annual International Symposium on Computer Architecture*, pages 426–436, 1983.

[16] Arvind and R. S. Nikhil. *Executing a Program on the MIT Tagged-Token Dataflow Architecture.* Computation Structures Group Memo 271, Laboratory for Computer Science, MIT, 1987.

[17] Arvind, R. S. Nikhil, and K. K. Pingali. I-Structures: Data structures for parallel computing. *TOPLAS*, 11(4):598–632, October 1989.

[18] Arvind and R.S. Nikhil. Executing a program on the MIT tagged-token dataflow architecture. *IEEE Transactions on Computers*, 39(3):300–318, March 1990.

[19] M. Babu et al. *An Enable Memory Controller Chip.* Technical Report, McGill University, November 1989. In the Proceedings of the VLSI Research Review, Centre de Recherche Informatique de Montréal.

[20] J. Backus. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641, August 1978.

[21] G. H. Barnes et al. The ILLIAC IV computer. *IEEE Transactions on Computer*, C-17(8), 1968.

[22] M. S. Bazaraa and J. J. Jarvis. *Linear Programming and Network Flows.* John Wiley and Sons Inc., 1977.

[23] Micah Beck, Keshav K. Pingali, and Alex Nicolau. *Static Scheduling for Dynamic Dataflow Machines.* Technical Report TR 90-1076, Dept. of Computer Science, Cornell University, Ithaca, NY, January 1990.

[24] G. A. Boughton. *Routing Networks in Packet Communication Architecture.* Master's thesis, Dept. of Electrical Engineering and Computer Science, MIT, June 1978.

[25] G. A. Boughton. *Routing Networks in Packet Communication Architecture.* PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, June 1985.

[26] W. J. Bouknight, S. A. Denenberg, McIntyre, J. M. Randall, A. H. Sameh, and D. L. Slotnick. The Illiac IV system. In *Proceedings of the IEEE*, 1972.

[27] J. D. Brock. Consistent semantics for a data flow language. In *Ninth International Symposium on Mathematical Fundations of Computer Science*, Poland, September 1980.

[28] J. D. Brock. *A Formal Model of Non-Determinate Dataflow Computation.* MIT/LCS/TR- 309, Laboratory for Computer Science, MIT, 1983.

[29] J. D. Brock. *Operational Semantics of a Data Flow Language.* Technical Report MIT/LCS/TM-120, Laboratory for Computer Science, MIT, 1978.

[30] J. Cocke. The search for performance in scientific processors. *Communications of the ACM*, 31(3), March 1988.

[31] Control Data Corp. *CDC Cyber 200 / Model 205 Technical Description.* St. Paul, Minnesota, 1980.

[32] D. E. Culler and Arvind. Resource requirements of dataflow programs. In *Proceedings of the 15th Annual International Symposium on Computer Architecture*, pages 141–150, 1988.

[33] G. Dantzig. Application of the simplex method to a transportation problem. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, John Wiley and Sons Inc., 1951.

[34] J. B. Dennis. Data flow for supercomputers. In *Proceeding of 1984 CompCon*, March 1984.

[35] J. B. Dennis. Data flow models of computation. In *Notes from lectures at the International Summer School on Control Flow and Data flow: Concepts of Distributed Programming*, Springer-Verlag, Marktoberdorf, Germany, 1984.

[36] J. B. Dennis. Data flow supercomputers. *IEEE Computer*, 13(11):48–56, November 1980.

[37] J. B. Dennis. Dataflow Computation: A Case Study. In Veljko Milutinović, editor, *Computer Architecture: Concepts and Systems*, pages 354–404, North-Holland, New York, 1988.

[38] J. B. Dennis. *First Version of a Data Flow Procedure Language.* Technical Report MIT/LCS/TM-61, Laboratory for Computer Science, MIT, 1975.

[39] J. B. Dennis. *High Speed Data Flow Computer Architecture for the Solution of Navier-Stokes Equations.* Computation Structures Group Memo 225, Laboratory for Computer Science, MIT, 1982.

[40] J. B. Dennis. Packet communication architecture. In *Proceedings of the 1975 Sagamore Computer Conference on Parallel Processing*, 1975.

[41] J. B. Dennis, G. A. Boughton, and Leung C. K. C. Building blocks for data flow prototypes. In *Proceedings of the Seventh Symposium on Computer Architecture*, May 1980.

[42] J. B. Dennis and J. B. Fossen. *Introduction to Data Flow Schemas.* Computation Structures Group Memo 81-1, Laboratory for Computer Science, MIT, Cambridge, MA, Sept. 1973.

[43] J. B. Dennis and G. R. Gao. An efficient pipelined dataflow processor architecture. In *Joint Conference on Supercomputing*, pages 368–373, IEEE Computer Society and ACM SIGARCH, Florida, November 1988.

[44] J. B. Dennis and G. R. Gao. Maximum pipelining of array operations on static data flow machine. In *Proceedings of the 1983 International Conference on Parallel Processing*, August 1983.

[45] J. B. Dennis, G. R. Gao, and K. Todd. *A Data Flow Supercomputer.* Computation Structures Group Memo 213, Laboratory for Computer Science, MIT, March 1982.

[46] J. B. Dennis, G. R. Gao, and K. W. Todd. Modeling the weather with a data flow supercomputer. *IEEE Transactions on Computers*, C-33(7):592–603, 1984.

[47] J. B. Dennis and D. P. Misunas. A preliminary architecture for a basic data-flow processor. In *The Second Annual Symposium on Computer Architecture*, pages 126–132, January 1975.

[48] J. B. Dennis, Y-P.L. Willie, and W. B. Ackerman. The MIT data flow engineering model. In *Proceedings of the IFIP 9th World Computer Congress*, Paris, France, September 1983.

[49] D. M. Dias and J. R. Jump. Analysis and simulation of buffered delta networks. *IEEE Transactions on Computers*, C-30(4), April 1981.

[50] D. M. Dias and J. R. Jump. Packet switching interconnection networks for modular systems. *IEEE Computer*, 14(12), December 1981.

[51] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerishe Mathematik*, 1:269–271, 1959.

[52] D. J. Evans. Parallel S.O.R. iterative methods. *Parallel Computing*, 1(1), August 1984.

[53] J. T. Feo. An analysis of the computational and parallel complexity of the Livermore Loops. *Parallel Computer*, 8(7):163–185, July 1988.

[54] L. R. Ford and D. R. Fulkerson. *Flow in Networks.* Princeton University Press, Princeton, NJ, 1962.

[55] D. D. Gajksi, D. A. Padua, D. J. Kuck, and R. H. Kuhn. A second opinion on data flow machines and languages. *IEEE Computer*, 15(2):58–69, February 1982.

[56] D. D. Gajski, D. J. Kuck, D. Lawrie, and A. Sameh. Cedar—a large scale multiprocessor. In *Proceeding of 1983 International Conference on Parallel Processing*, August 1983.

[57] G. R. Gao. *A Flexible Architecture Model for Hybrid Dataflow and Control-Flow Evaluation.* ACAPS Technical Memo 07, School of Computer Science, McGill University, Montreal, Que., January 1989.

[58] G. R. Gao. A flexible architecture model for hybrid dataflow and control-flow evaluation. In *Proceedings of the International Workshop: Dataflow—A Status Report*, in conjunction with the ACM Annual Symposium on Computer Architecture, Jerusalem, Israel, May 1989. To be published by Prentice-Hall.

[59] G. R. Gao. Homogeneous approach of mapping data flow programs. In *Proceedings of the 1983 International Conference on Parallel Processing*, August 1984.

[60] G. R. Gao. *An Implementation Scheme for Array Operations in Static Data Flow Computer.* Master's thesis, Laboratory of Computer Science, MIT, Cambridge, MA, June 1982.

[61] G. R. Gao. Massive fine-grain parallelism in array computation—a data flow solution. In *Proceedings of Future Directions of Supercomputer Architecture and Software*, Charleston, SC, May 1986.

[62] G. R. Gao. A maximally pipelined tridiagonal linear equation solver. *Journal of Parallel and Distributed Computing*, 3(2):215–235, June 1986.

[63] G. R. Gao. Maximally pipelining linear recurrence on static dataflow computers. *International Journal of Parallel Programming*, 15(2):127–149, April 1987.

[64] G. R. Gao. *Maximum Pipelining of Linear Recurrence on Static Data Flow Computers.* Computation Structures Group Memo 49, Laboratory for Computer Science, MIT, August 1985.

[65] G. R. Gao. A stability classification method and its application to pipelined solution of linear recurrences. *Parallel Computing*, 305–321, April 1987.

[66] G. R. Gao and K. Theobald. *An Enable Memory Controller Chip for a Static Data Flow Computer.* CSG Design Note 18, Laboratory for Computer Science, MIT, January 1985.

[67] G. R. Gao and R. Tio. Instruction set design of an efficient pipelined dataflow architecture. In *Proceedings of the 22nd International Conference of System Science*, pages 383–393, Hawaii, January 1989.

[68] G. R. Gao, Robert Kim Yates, and Lenore Restifo Mullin. *An Efficient Monolithic Array Constructor.* ACAPS Technical Memo 19, School of Computer Science, McGill University, Montreal, Que., June 1990. To appear in *Proceedings of the Third Workshop on Programming Languages and Compilers for Parallel Computing, Irvine, CA, August 1-3, 1990*, MIT Press.

[69] J. R. Gurd, C. C. Kirkham, and I. Watson. The Manchester prototype dataflow computer. *Communications of the ACM*, 28(1):34–52, January 1985.

[70] R. W. Hockney and C. R. Jesshope. *Parallel Computers 2: Architecture, Programming and Algorithms.* Adam Hilger Ltd., Bristol, 1988.

[71] W.-K. Hong. *IF1 Parser for HDDG.* ACAPS Design Note 01, School Of Computer Science, McGill University, Montreal, Que., June 1988.

[72] K. Hwang and F. A. Briggs. *Computer Architecture and Parallel Processing.* McGraw Hill Book Company, New York, 1984.

[73] K. Hwang and S. P. Su. *Multitask Scheduling in Vector Supercomputers.* TR-EE 83–52, School of Electrical Engineering, Purdue University, December 1983.

[74] R. A. Iannucci. Toward a dataflow/von Neumann hybrid architecture. In *Proceedings of the 15th Annual International Symposium on Computer Architecture*, pages 131–140, 1988.

[75] P. A. Jensen and J. W. Barnes. *Network Flow Programming.* John Wiley and Sons Inc., 1980.

[76] R. M. Karp and R. E. Miller. Properties of a model for parallel computation: determinancy, termination, queueing. *SIAM Journal of Applied Math.*, 14, November 1966.

[77] R. M. Keller, G. Lindstrom, and S. Patil. A loosely-coupled applicative multi-processing system. In *AFIPS Conference Proceedings, vol. 48*, pages 613–622, 1979.

[78] L. G. Khachian. A polynomial algoritm in linear programming. *Soviet Math. Doklady*, 20:191–194, 1979.

[79] D. E. Knuth. *The Art of Computer Programming. Second Edition, Vol 1: Fundamental Algorithms*, Addison-Wesley, 1968.

[80] P. M. Kogge. *The Architecture of Pipelined Computers.* McGraw-Hill Book Company, New York, 1981.

[81] P. M. Kogge. Maximum rate pipelined solutions to recurrence problems. In *Proceedings of the 1st Computer Architecture Symposium*, pages 71–76, December 1973.

[82] P. M. Kogge. A parallel algorithm for efficient solution of a general class of recurrence equations. *IEEE Transactions on Computers*, C-22(8), August 1973.

[83] P. M. Kogge. Parallel solutions of recurrence problems. *IBM Journal of Research and Development*, 18(2), March 1974.

[84] C. P. Kruskal and M. Snir. The performance of multistage interconnection networks. *IEEE Transactions on Computers*, C-32(12), December 1983.

[85] D. J. Kuck. A survey of parallel machine organization and programming. *Computing Surveys*, 9(1), March 1977.

[86] D. J. Kuck, R. H. Kuhn, D. A. Padua, B. Leasure, and M. Wolfe. Analysis and transformation of programs for parallel computation.

In *Proceedings of the Fourth International Computer Software and Application Conference*, October 1980.

[87] D. J. Kuck, R. H. Kuhn, D. A. Padua, B. Leasure, and M. Wolfe. Dependence graphs and compiler optimizations. In *Proceedings of the 8th ACM Symposium on Principles of Programming Languages*, pages 207–218, 1981.

[88] D. J. Kuck, Y. Muraoka, and S. C. Chen. On the number of operations simultaneously executable in FORTRAN-like programs and their resulting speed-up. *IEEE Transactions on Computers*, C-21(12), December 1972.

[89] H. T. Kung. Why systolic architecture. *Computer*, 15(1), January 1982.

[90] E. Lawler. *Combinatorial Optimization Networks and Matroids.* Holt, Rinehart, and Winston, 1976.

[91] T. Leighton. *Parallel Computation Using Meshes of Trees (Extended Abstract).* Technical Report, Mathematics Dept. and Laboratory for Computer Science, MIT, 1981.

[92] C. E. Leiserson. *Area Efficient VLSI Computation.* PhD thesis, Dept of Computer Science, CMU, October 1981.

[93] C. E. Leiserson. *Area-Efficient VLSI Computation.* MIT Press, Cambridge, MA, 1983.

[94] C. E. Leiserson. *Optimizing Synchronous Systems.* Technical Memo 215, Laboratory for Computer Science, MIT, 1982.

[95] J. R. McGraw. The VAL language: Decription and analysis. *ACM TOPLAS*, 4(1):44–82, January 1982.

[96] J. R. McGraw, S. Skedzielewski, et al. *SISAL: Streams and Iteration in a Single Assignment Language—Language Reference Manual Version 1.2.* Technical Report M-146, Lawrence Livermore National Laboratory, 1985.

[97] F. H. McMahon. *Livermore FORTRAN Kernels: A Computer Test of the Numerical Performance Range.* Technical Report UCRL-53745, Lawrence Livermore National Laboratory, Livermore, CA, December 1986.

[98] G. Milne and R. Milner. Concurrent processes and their syntax. *Journal of the ACM*, 26(2), April 1979.

[99] R. Milner. Flowgraphs and flow algebras. *Journal of ACM*, 26(4), October 1979.

[100] K. Miura and K. Uchida. FACOM vector processor VP-100/VP-200. In *Proceedings of the Nato Advanced Research Workshop on High Speed Computing*, Springler-Verlag, Julich, W. Germany, 1983.

[101] J. J. Moder and C. R. Phillips. *Project Management with CPM and PERT*. Van Nostrand Reinfold Co., New York, 1970.

[102] L. B. Montz. *Safety and Optimization Transformations for Data Flow Programs*. Technical Report 240, Laboratory for Computer Science, MIT, Cambridge, MA, January 1980.

[103] R. Nikhil and Arvind. Can dataflow subsume von Neumann computing? In *Proceedings of the 16th International Symposium on Computer Architecture*, pages 262–272, Israel, 1989.

[104] D. A. Padua, D. J. Kuck, and D. L. Lawrie. High speed multiprocessor and compilation techniques. *IEEE Transactions on Computers*, C–29(9):763–776, September 1980.

[105] G. M. Papadopoulos and D. E. Culler. Monsoon: an explicit token-store architecture. In *Proceedings of the Seventeenth Annual International Symposium of Computer Architecture, Seattle, Washington*, pages 82–91, 1990.

[106] Z. Paraskevas. *Code Generation for Dataflow Software Pipelining*. Master's thesis, McGill University, Montreal, Quebec, June 1989.

[107] Z. Paraskevas. *SISAL-Kernel: An Experimental Language for the Argument-Fetching Architecture*. ACAPS Technical Note 12, School of Computer Science, McGill University, Montreal, Que., June 1989.

[108] S. S. Patil. Closure properties of interconnections of determinate systems. In *The project MAC Conference on Concurrent Systems and Parallel Computation*, pages 107–116, ACM, 1970.

[109] J. E. Rodriguez. *A Graph Model for Parallel Computation.* PhD thesis, Laboratory for Computer Science, MIT, Cambridge, MA, 1969.

[110] C. A. Ruggiero and J. Sargeant. Control of parallelism in the Manchester dataflow machine. In *Functional Programming Languages and Computer Architecture*, pages 1–15, Springer-Verlag, LNCS-274, 1987.

[111] R. M. Russel. The Cray-1 computer system. *Communications of the ACM*, 21(1), January 1978.

[112] S. Sakai et al. An architecture of a dataflow single chip processor. In *Proceedings of the 16th International Symposium on Computer Architecture*, pages 46–53, Israel, 1989.

[113] Stephen K. Skedzielewski and John Glauert. *IF1: An Intermediate Form for Applicative Languages.* Technical Report M-170, Version 1.0, Lawrence Livermore National Laboratory, July 1985.

[114] H. Stone. Parallel tridiagonal equation solvers. *ACM Transactions on Mathematical Software*, 1, 1975.

[115] R. Tio. *The A-Code Assembly Language Reference Manual.* ACAPS Design Note 02, School Of Computer Science, McGill University, Montreal, Que., July 1988.

[116] R. Tio. *DASM: The A-Code Data-Driven Assembler Program Reference Manual.* ACAPS Design Note 03, School Of Computer Science, McGill University, Montreal, Que., July 1988.

[117] K. W. Todd. Function sharing in a static data flow machine. In *Proceedings of the 1982 International Conference on Parallel Processing*, pages 137–139, 1982.

[118] K. W. Todd. *High Level VAL Constructs in a Static Data Flow Machine.* Technical Report 262, Laboratory for Computer Science, MIT, 1981.

[119] K. W. Todd. *An Interpreter for Instruction Cells.* Computation Structures Group Memo 208, Laboratory for Computer Science, MIT, 1982.

[120] G. Tremblay and G. R. Gao. *A Formal Operational Model for Data-Driven Program Tuples.* ACAPS Technical Memo 08, School of Computer Science, McGill University, Montreal, Que., April 1989.

[121] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proceedings of the 13th Annual Symposium on Theory of Computing*, May 1981.

[122] I. Watson and J. Gurd. A practical data flow computer. *IEEE Computer*, 15(2):51–57, February 1982.

[123] K.-S. Weng. *An Abstract Implementation for a Generalized Data Flow Language.* Technical Report MIT/LCS-TR-228, Laboratory for Computer Science, MIT, 1979.

[124] T. Yuba et al. Sigma-1: a dataflow computer for scientific computations. *Computer Physics Communications*, 37:141–148, 1985.

Index