References

1. The national technology roadmap for semiconductors. Semiconductor Industry Association, 1999. Available from: SEMATECH, 3101 Industrial Terrace Suite 106 Austin TX 78758.
2. Handel-C language datasheet. Available from Celoxica Ltd: `http://www.celoxica.com/`.
3. Haskell98 report. Available from `http://www.haskell.org/`.
4. PHP hypertext preprocessor. See `http://www.php.net/`.
5. Afred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques and Tools*. Addison Wesley, 1986.
6. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
7. Marco Aldinucci. The meta transformation tool for skeleton-based languages. In *Proceedings of the 2nd International Workshop on Constructive Methods for Parallel Programming (CMPP)*, 2000. Available from: `http://citeseer.nj.nec.com/486282.html`.
8. J. Aldrich, C. Chambers, E. Sirer, and S. Eggers. Static analyses for eliminating unnecessary synchronization from java programs. In *Proceedings of the International Symposium on Static Analysis*, volume 1694 of *LNCS*. Springer-Verlag, 1999.
9. American National Standards Institute, Inc. *The Programming Language ADA Reference Manual*. Springer-Verlag, 1983.
10. A. Appel. *Modern Compiler Implementation in Java/ML/C*. Cambridge University Press, 1998.
11. A. W. Appel and D. B. MacQueen. Standard ML of New Jersey. In J. Maluszyński and M. Wirsing, editors, *Proceedings of the Third International Symposium on Programming Language Implementation and Logic Programming*, number 528, pages 1–13. Springer-Verlag, 1991.
12. Arvind and Xiaowei Shen. Using term rewriting systems to design and verify processors. *IEEE Micro (Special Issue on Modeling and Validation of Microprocessors*, May/June 1999.
13. B. Preas and M. Lorenzetti. *Physical Design Automation of VLSI-Systems*. Benjamin Cummings, 1989.
14. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

15. J. Backus. Can functional programming be liberated from the von Neumann style? *Communications of the ACM*, 21(8):613–641, 1978.
16. J. Backus. The algebra of functional programs: Function level reasoning, linear equations and extended definitions. In *Proceedings of the Symposium on Functional Languages and Computer Architecture*, June 1981.
17. F. Balarin, M. Chiodo, P. Giusto, H. Hsieh, A. Jurecska, L. Lavagno, C. Passerone, A. Sangiovanni-Vincentelli, E. Sentovich, K. Suzuki, and B. Tabbara. *Hardware-Software Co-Design of Embedded Systems: The Polis Approach*. Kluwer Academic Press, June 1997.
18. Mario R. Barbacci and Daniel P. Siewiorek. Automated exploration of the design space for register transfer (RT) systems. In *Proceedings of the 1st Annual Symposium on Computer Architecture (ISCA)*, pages 101–106, 1973.
19. A. Bardsley and D. A. Edwards. The Balsa asynchronous circuit synthesis system. In *Proceedings of the Forum on Design Languages*, 2000. Available on request from European Electronic Chips and Systems design Initiative (ECSI).
20. G. Berry. Real time programming: special purpose or general purpose langages. Technical Report RR-1065, INRIA, August 1989.
21. G. Berry. A hardware implementation of pure esterel. *SADHANA – Academy Proceedings in Engineering Sciences*, 17:95–130, March 1992.
22. G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.
23. G. Berry and G. Gonthier. The Esterel synchronous programming language: design, semantics, implementation. Technical Report 842, INRIA, 1988.
24. G. Berry, S. Ramesh, and R. K. Shyamasundar. Communicating reactive processes. In *Conference Record of the Twentieth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 85–98. Charleston, South Carolina, 1993.
25. P. Bjesse, K. Claessen, M. Sheeran, and S. Singh. Lava: Hardware description in Haskell. In *Proceedings of the 3rd International Conference on Functional Programming*, SIGPLAN. ACM, 1998.
26. B. Bose. DDD: A transformation system for digital design derivation. Technical Report 331, Indiana University, 1991.
27. B. Bose. *DDD-FM9001: Derivation of a Verified Microprocessor*. PhD thesis, Indiana University, 1994.
28. R. Brayton, R. Camposano, G. De Micheli, R. Otten, and J. van Eijndhoven. *The Yorktown Silicon Compiler System*. Addison-Wesley, 1988.
29. R.M. Burstall and J. Darlington. A transformation system for developing recursive programs. In *JACM 24(1)*, 1977.
30. L. Cardelli. The functional abstract machine. Technical Report TR-107, AT&T Bell Laboratories, April 1983.
31. C.A.R.Hoare. *Communicating Sequential Processes*. Prentice-Hall International, 1985.
32. R. Cartwright and M. Fagan. Soft typing. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, SIGPLAN. ACM, 1991.
33. Gregory J. Chaitin. Register allocation spilling via graph coloring. In *SIGPLAN Symposium on Compiler Construction*, pages 98–105, 1982.
34. D. Chapiro. *Globally-Asynchronous Locally-Synchronous systems*. PhD thesis, University of Stanford, 1984.

35. P. Chou, R. Ortega, and G. Borriello. The Chinook hardware/software co-synthesis system. In *Proceedings of the 8th International Symposium on System Synthesis*, 1995.
36. C. Clack and S. L. Peyton-Jones. Strictness analysis – a practical approach. In *Functional Languages and Computer Architecture*, LNCS, pages 35–49. Springer-Verlag, 1985.
37. Koen Claessen and David Sands. Observable sharing for functional circuit description. In *Asian Computing Science Conference*, pages 62–73, 1999.
38. Koen Classen, Mary Sheeran, and Stanam Singh. The design and verification of a sorter core. In *Proceedings of the 11th Advanced Working Conference on Correct Hardware Design and Verification Methods*, volume 2144 of *LNCS*, pages 355–369. Springer-Verlag, 2001.
39. B. Cook, J. Launchbury, and J. Matthews. Specifying superscalar microprocessors with Hawk. In *Proceedings of the workshop on formal techniques for hardware*, June 1998.
40. A. Davis and S. M. Nowick. An introduction to asynchronous circuit design. Technical Report UUCS-97-013, University of Utah, September 1997.
41. G. De Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill Inc., 1994.
42. G. De Micheli and D. Ku. HERCULES - a system for high-level synthesis. In *Proceedings of the Design Automation Conference*, pages 483–488. ACM Press, June 1988.
43. G. De Micheli, D. Ku, F. Mailhot, and T. Truong. The Olympus synthesis system for digital design. *Design & Test of Computers*, October 1990.
44. D. Edwards and A. Bardsley. Balsa 3.0 user manual. Available from `http://www.cs.man.ac.uk/amulet/projects/balsa/`.
45. C. Famsworth, D.A. Edwards, J. Liu, and S.S. Sikand. A hybrid asynchronous system design environment. In *Proceedings of the Second Working Conference on Asynchronous Design Methodologies (ASYNC 95)*. IEEE.
46. Martin Feather. *A system for developing programs by transformation*. PhD thesis, University of Edinburgh, 1979.
47. Jeanne Ferrante, Karl J. Ottenstein, and Joe D. Warren. The program dependence graph and its use in optimization. *ACM Transactions on Programming Languages and Systems*, 9(3):319–349, July 1987.
48. Simon Frankau and Alan Mycroft. Stream processing hardware from functional language specifications. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS)*. IEEE Computer Society Press, January 2003.
49. Simon Frankau, Alan Mycroft, and Simon Moore. Statically-allocated languages for hardware stream processing (extended abstract). In Sambuddhi Hettiaratchi, editor, *UK ACM SIGDA Workshop on Electronic Design Automation*. UK ACM SIGDA, Bournemouth University, September 2002.
50. T.D. Friedman and S.C. Yang. Methods used in an automatic logic design generator (ALERT). *IEEE Transactions in Computing*, C-18:593–614, 1969.
51. S. Furber, D. Edwards, and J. Garside. AMULET3: a 100 MIPS asynchronous embedded processor. pages 329–334. IEEE, 2000.
52. Daniel Gajski, T. Ishii, V. Chaiyukul, H. Juan, and T. Hadley. A design methodology and environment for interactive behavioural synthesis. Technical Report 96-26, Department of Information and Computer Science, University of California, Irvine, June 1996.

53. D.D. Gajski and L. Ramachandran. Introduction to high-level synthesis. *Design & Test of Computers*, 11(4), 1994.
54. Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software Practice and Experience*, 30(11):1203–1233, September 2000. ISSN 0038-0644.
55. E.F. Girczcy. *Automatic Generation of Microsequenced Data Paths to Realize ADA Circuit Descriptions*. PhD thesis, Carleton University, Ottowa, Canada, July 1984.
56. Lance Glasser and Daniel Dobberpuhl. *The Design and Analysis of VLSI Circuits*. Addison-Wesley, 1985.
57. C.K. Gomard and P. Sestoft. Globalization and live variables. In *Proceedings of the Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, pages 166–177. ACM Press, 1991.
58. P. Le Guernic, M. Le Borgne, T. Gautier, and C. Le maire. Programming real time applications with Signal. *Proceedings of the IEEE*, 79:1321–1336, September 1991.
59. S. Guo and W. Luk. Compiling Ruby into FPGAs. In *Field Programmable Logic and Applications*, volume 975 of *LNCS*, pages 188–197. Springer-Verlag, 1995.
60. N. Halbwachs. *Synchronous programming of reactive systems*. Kluwer Academic Press, 1993.
61. N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud. The synchronous data flow programming language Lustre. *Proceedings of the IEEE*, 79(9):1305–1321, September 1991.
62. N. Halbwachs, A. Lonchampt, and D. Pilaud. Describing and designing circuits by means of the synchronous data-flow programming language Lustre. In IFIP Working Conference: From HDL Descriptions to Guaranteed Correct Circuit Designs, Grenoble, September 1986.
63. K. Hammond. Parallel functional programming: An introduction. In *Proceedings of the International Symposium on Parallel Symbolic Computation*. World Scientific, 1994.
64. D. Harel and A. Pnueli. On the development of reactive systems. In K.R.Apt, editor, *Logics and Models of Concurrent Systems*, volume F-13 of *NATO ASI*. Springer-Verlag, 1985.
65. David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.
66. Paul Havlak. Construction of thinned gated single-assignment form. In *1993 Workshop on Languages and Compilers for Parallel Computing*, number 768 in LNCS, pages 477–499. Springer-Verlag, 1993.
67. J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers Inc., 1990.
68. James Hoe, Martin Rinard, and Arvind. An exercise in high-level architectural description using a synthesizable subset of term rewriting systems, 1997. Computation Structures Group Memo 403.
69. J.C. Hoe and Arvind. Hardware synthesis from term rewriting systems. In *Proceedings of X IFIP International Conference on VLSI*, 1999.
70. M. Hofmann. A type system for bounded space and functional in-place update. In *Proceedings of the 9th European Symposium On Programming*, LNCS. Springer-Verlag, 2000.
71. J. Hughes and L. Pareto. Recursion and dynamic data-structures in bounded space: Towards embedded ML programming. *Proceedings of the International Conference on Functional Programming (ICFP)*, 34(9):70–81, 1999.

72. John Hughes, Lars Pareto, and Amr Sabry. Proving the correctness of reactive systems using sized types. In *Symposium on Principles of Programming Languages*, pages 410–423, 1996.
73. IEEE. Standard VHDL Reference Manual, 1993. IEEE Standard 1076-1993.
74. IEEE. Verilog HDL language reference manual. IEEE Draft Standard 1364, October 1995.
75. IEEE. Standard for VHDL Register Transfer Level (RTL) Synthesis, 1999. IEEE Standard 1076.6-1999.
76. Inmos (Ltd.). *Occam 2 Reference Manual*. Prentice Hall, 1998.
77. Wolfgang Fichtner Jens Muttersbach, Thomas Villiger. Practical design of globally-asynchronous locally-synchronous systems. In *6th International Symposium on Advanced Research in Asynchronous Circuits and Systems*. IEEE Press, 2000.
78. E.L. Johnson and G.L. Nemhauser M.W.P. Savelsbergh. Progress in integer linear programming: an exposition. Technical Report LEC-97-02, Georgia Institute of Technology, School of Industrial and Systems Engineering, January 1997.
79. S.D. Johnson and B. Bose. DDD: A system for mechanized digital design derivation. Technical Report 323, Indiana University, 1990.
80. G. Jones and M. Sheeran. Circuit design in Ruby. In J. Staunstrup, editor, *Formal Methods for VLSI design*, pages 13–70. North-Holland, 1990.
81. N. Jones, C. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall, 1993.
82. Jens-Peter Kaps and Christof Paar. Fast DES implementation for FPGAs and its application to a universal key-search machine. In *Selected Areas in Cryptography*, volume 1556 of *Lecture Notes in Computer Science*, pages 234–247. Springer-Verlag, 1998. URL `citeseer.nj.nec.com/119314.html`.
83. B. Kernighan and D. Ritchie. *The C Programming Language. Second Edition*. Prentice Hall, 1988.
84. Joep Kessels, Kees van Berkel, Ronan Burgess, Marly Roncken, and Frits Schalij. An error decoder for the compact disc player as an example of VLSI programming. In *Proc. European Conference on Design Automation (EDAC)*, pages 69–74. IEEE, 1992.
85. Anne T. Kohlstaedt. Daisy 1.0 reference manual. Technical Report 119, Indiana University Computer Science Department, 1981.
86. D. Ku and G. De Micheli. HardwareC—a language for hardware design (version 2.0). Technical Report CSL-TR-90-419, Stanford University, 1990.
87. D. Ku and G. De Micheli. High-level synthesis and optimization strategies in Hercules and Hebe. In *Proceedings of the European ASIC Conference*, pages 124–129, May 1990.
88. D. Ku and G. De Micheli. Constrained resource sharing and conflict resolution in Hebe. *Integration—The VLSI Journal*, December 1991.
89. D. Ku and G. De Micheli. Relative scheduling under timing constraints: Algorithm for high-level synthesis of digital circuits. *Transactions on CAD/ICAS*, pages 697–718, June 1992.
90. P. Landin. The mechanical evaluation of expressions. *The Computer Journal*, 6 (4):308–320, 1964.
91. R. Lipsett, C. Schaefer, and C. Ussery. *VHDL: Hardware Description and Design*. Kluwer Academic Publishers, Boston, MA, 1992.
92. H. De Man, J. Rabaey, P. Six, and L. Claesen. Cathedral-II: A silicon compiler for digital signal processing. *Design & Test of Computers*, December 1986.

93. G. Marchioro, J. Daveau, and A. Jerraya. Transformation partitioning for co-design of multiprocessor systems. In *Proccedings of the International Conference on Computer Aided Design (ICCAD)*. IEEE, 1997.
94. S. Marlow, S. Peyton Jones, A. Moran, and J. Reppy. Asynchronous exceptions in Haskell. In *Proceedings of the Conference on Programming Language Design and Implementation (PLDI)*, SIGPLAN. ACM, 2001.
95. P. Marwedel. A new synthesis algorithm for the mimola software system. In *Proceedings of the 23rd Design Automation Conference*, pages 271–277. ACM Press, 1986.
96. J. Matthews, B. Cook, and J. Launchbury. Microprocessor specification in Hawk. In *Proceedings of the IEEE International Conference on Computer Languages*, 1998.
97. M. McFarland. Using bottom-up design techniques in the synthesis of hardware from abstract behavioral descriptions. In *Proceedings of the 23rd Design Automation Conference*, pages 474–480. ACM Press, 1986.
98. M. McFarland, A. Parker, and R. Camposano. The high-level synthesis of digital systems. *Proceedings of the IEEE*, 78(2), February 1990.
99. R. Milner. A theory of type-polymorphism in programming. *Journal of Computer and System Sciences*, 17(3), 1978.
100. R. Milner. The polyadic π-calculus: A tutorial. Technical Report ECS-LFCS-91-180, University of Edinburgh, October 1991.
101. R. Milner, M. Tofte, R. Harper, and D. MacQueen. *The Definition of Standard ML (Revised)*. MIT Press, 1997.
102. S.W. Moore, G.S. Taylor, P.A. Cunningham, R.D. Mullins, and P.Robinson. Using stoppable clocks to safely interface asynchronous and synchronous subsystems. In *Proceedings of the AINT (Asynchronous INTerfaces) Workshop*, July 2000.
103. Steven S. Muchnick. *Advanced Compiler Design and Implementation*. Morgan Kaufmann Publishers, San Francisco, CA, 1997.
104. A. Mycroft and R. Sharp. Higher-level techniques for hardware description and synthesis. *Software Tools for Technology Transfer (STTT)*. To Appear.
105. A. Mycroft and R.W. Sharp. A statically allocated parallel functional language. In *Proceedings of the International Conference on Automata, Languages and Programming*, volume 1853 of *LNCS*. Springer-Verlag, 2000.
106. A. Mycroft and R.W. Sharp. Hardware/software co-design using functional languages. In *Proceedings of TACAS*, volume 2031 of *LNCS*. Springer-Verlag, 2001.
107. T. Nijhar and A. Brown. Source-level optimisation of VHDL for behavioural synthesis. *Proceedings on Computers and Digital Techniques*, 144(1):1–6, January 1997.
108. Rishiyur Nikhil and Arvind. *Implicit Parallel Programming in pH*. Morgan Kaufmann, May 2001.
109. John O'Donnell. Hardware description with recursion equations. In *Proceedings of the IFIP 8th International Symposium on Computer Hardware Description Languages and their Applications*, pages 363–382. North-Holland, April 1987.
110. John O'Donnell. Generating netlists from executable circuit specifications in a pure functional language. In *Functional Programming, Workshops in Computing, Proceedings*, pages 178–194. Springer-Verlag, 1992.
111. S. Olcoz and J. Colom. Towards a formal semantics of IEEE std. VHDL 1076. In *Proceedings of 1993 European Design Automation Conference with Euro-VHDL*. IEEE, 1993.
112. I. Page. Compiling Occam into Field-Programmable Gate Arrays. In Moore and Luk, editors, *FPGAs*, pages 271–283. Abingdon EE&CS Books, 1991.

113. I. Page. Parameterised processor generation. In Moore and Luk, editors, *More FPGAs*, pages 225–237. Abingdon EE&CS Books, 1993.
114. I. Page. Reconfigurable processor architectures. *Microprocessors and Microsystems*, 20(3):185–196, May 1996.
115. Samir Palnitkar. *Verilog HDL: A guide to digital design and synthesis*. Prentice Hall, 1996. ISBN 0-13-451675-3.
116. B. M. Pangrle. Splicer: A heuristic approach to connectivity binding. In *Proccedings of the 25th Design Automation Conference*, pages 536–541. ACM Press, June 1988.
117. N. Park and A. Parker. Sehwa: A software package for synthesis of pipelines from behavioral specifications. *IEEE Transactions on Computer-Aided Design*, pages 356–370, March 1988.
118. Helmuth Partsch, Wolfram Schulte, and Ton Vullinghs. System support for the interactive transformation of functional programs. In *Proceedings of the 5th Workshop on Tools for System Design and Verification (FM-TOOLS)*, 2002. Available from `http://citeseer.nj.nec.com/336476.html`.
119. P. Paulin and J. Knight. Force-directed scheduling for the behavioral synthesis of ASICs. *IEEE transactions on Computer-Aided Design*, 6:661–679, July 1989.
120. Lawrence Paulson. *ML for the working programmer*. Cambridge University Press, 1996.
121. Ad Peeters. Re: Tangram and balsa. Personal communication by email. (Peeters is a Senior Scientist at Philips Research Eindhoven).
122. G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.
123. S. Pommier. *An Introduction to APL*. Cambridge University Press, 1983.
124. R. Chapman and Deok-Hyun Hwang. A process-algebraic semantics for VHDL. In W. Ecker, editor, *SIG-VHDL Spring '96 Working Conference*, pages 157–168. Shaker Verlag, Dresden, Germany, 1996.
125. Jonathan Rees, W. Clinger, et al. The revised report 3 on the algorithmic language SCHEME. *SIGPLAN Notices*, 21(12):37–79, 1986.
126. S. Renault, A. Pettorossi, and M. Proietti. Design, implementation, and use of the MAP transformation system. Technical Report R. 491, IASI-CNR, Roma, Italy, December 1998.
127. Richard L. Rudell. Tutorial: Design of a logic synthesis system. In *Proceedings of the Design Automation Conference*, pages 191–196. ACM Press, 1996.
128. V. Sarkar. A concurrent execution semantics for parallel program graphs and program dependence graphs. In U. Banerjee, D. Gelernter, A. Nicolau, and D. Padua, editors, *Proceedings of the 5th International Workshop in Languages and Compilers for Parallel Computing*, volume 757 of *LNCS*, pages 16–30. Springer-Verlag, 1992.
129. V. Sarkar and B. Simons. Parallel program graphs and their classification. In *Proceedings of the Sixth Workshop on Languages and Compilers for Parallel Computing*, volume 768 of *LNCS*, pages 633–655. Springer-Verlag, 1993.
130. Bruce Schneier. *Applied cryptography: protocols, algorithms, and sourcecode in C*. John Wiley and Sons, New York, 1994. ISBN 0-471-59756-2.
131. R.W. Sharp and A. Mycroft. A higher-level language for hardware synthesis. In *Proceedings of the 11th Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, volume 2144 of *LNCS*. Springer-Verlag, 2001.
132. M. Sheeran. muFP, a language for VLSI design. In *Proceedings of the ACM Symposium on LISP and Functional Programming*, 1984.

133. D. Springer and D. Thomas. Exploiting the special structure of conflict and compatibility graphs in high-level synthesis. In *Proceedings of the International Conference on Computer Aided Design*, pages 254–257, 1990.
134. Tony Ma Stephen A. Edwards and Robert Damiano. Using a hardware model checker to verify software. In *Proceedings of the 4th International Conference on ASIC (ASICON)*. IEEE Press, 2001.
135. Donald E. Thomas, E. M. Dirkes, Robert A. Walker, Jayanth V. Rajan, J. A. Nestor, and Robert L. Blackburn. The system architect's workbench. In *Proceedings of the 25th Design Automation Conference*, pages 337–343. ACM Press, 1988.
136. C. Tseng and D.P. Siewiorek. Automated synthesis of data paths in digital systems. *IEEE Transactions on Computer-Aided Design*, 5(3), July 1986.
137. C. Van Berkel. *Handshake circuits: An Intermediary Between Communicating Processes and VLSI*. PhD thesis, Eindhoven University of Technology, 1992.
138. K. Van Berkel. *Handshake Circuits: an Asynchronous Architecture for VLSI Programming*, volume 5 of *International Series on Parallel Computation*. Cambridge University Press, 1993.
139. Hans van Gageldonk, Daniel Baumann, Kees van Berkel, Daniel Gloor, Ad Peeters, and Gerhard Stegmann. An asynchronous low-power 80C51 microcontroller. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 96–107, 1998.
140. J. van Tassel. *Femto-VHDL: The Semantics of a Subset of VHDL and its Embedding in the HOL Proof Assistant*. PhD thesis, University of Cambridge, 1992.
141. P. Wadler. Monads for functional programming. In *Advanced Functional Programming*, volume 925 of *LNCS*. Springer-Verlag, 1995.
142. R. Walker and D. Thomas. Behavioral transformations for algorithmic level IC design. *IEEE Transactons on Computer-Aided Design*, 8(10):1115–1127, 1989.
143. Neil Weste and Kamran Eshraghian. *Principles of CMOS VLSI design: A systems perspective (second edition)*. Addison-Wesley, 1993.
144. Philip A. Wilsey. Developing a formal semantic definition of VHDL. In Jean Mermet, editor, *VHDL for Simulation, Synthesis and Formal Proofs of Hardware*, pages 245–256. Kluwer Academic Publishers, 1992.
145. Z. Zhu and S.D. Johnson. An example of interactive hardware transformation. Technical Report TR383, Computer Science Department, Indiana University, 1993.
146. G. Zimmerman. Eine Methode zum Entwurf von Digitalrechnern mit der Programmiersprache MIMOLA. *Informatik-Fachberichte*, 5, 1976.

Index