References

Appel, A., Ellis, J. & Li, K. (1988) Fast garbage collection on stock multiprocessors. In *Proceedings of the Conference on Programming Language Design and Implementation, Atlanta*, June.

Augusteijn, A. & van der Hoeven, G. (1984) Combinatorgraphs as self-reducing programs. University of Twente, the Netherlands.

Augustsson, L. (1984) A compiler for lazy ML. In *Proceedings of the ACM Symposium on Lisp and Functional Programming, Austin*, pp. 218-27, August.

Backus, J. (1978) 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.

Baker, H. (1978) List processing in real time on a serial computer. *Communications of the ACM*, **21**(4):280-294, June.

Barendregt, H. P. (1981) *The Lambda Calculus: Its Syntax and Semantics*, Elsevier, New York.

Belinfante J. G. F. (1987) S/K/ID: Combinators in Forth. *Journal of Forth Application and Research*, **4**(4):555-580.

Bell, J. (1973) Threaded code. *Communications of the ACM*, **16**(6):370-372, June.

Burley, R. (1987) An overview of the four systems in the VAX 8800 family. *Digital Technical Journal*, **4**:10-19, February.

Clarke, T. J. W., Gladstone, P. J. S., MacLean, C. D. & Norman, A. C. (1980) SKIM - The S, K, I reduction machine. In *Conference Record of the 1980 LISP Conference, Stanford*, ACM, pp. 128-135, August.

Connolly, J. (1987) High-octane Microvaxes roll, *Computerworld*, **21**(37):1, September 14.

Curry, H. B. & Feys, R. (1968) *Combinatory Logic, Volume 1*, North-Holland.

Digital Equipment Corporation (1989) *DECstation 3100 Technical Overview (EZ-J4052-28)*, Digital Equipment Corporation, Maynard Massachusetts.

Fairbairn, J. & Wray, S. (1987) TIM: A simple, lazy abstract machine to execute supercombinators. In Kahn, G. (ed.), *Proceedings of the Conference on Functional Programming and Computer Architecture, Portland*, Springer Verlag, pp. 34-45.

Field, A. J. & Harrison, P. G. (1988) *Functional Programming*, Addison-Wesley, Wokingham England.

Friedman, D. P. & Wise, D. S. (1976) CONS should not evaluate its arguments. In Michaelson, S. & Milner, R. (eds.) *Proceedings of the 3rd ICALP conference, Edinburgh, Scotland*, pp. 257-284.

Harris Semiconductor (1989) *RTX 2000 Real Time Express Microcontroller Data Sheet*, Harris Corporation, Melbourne FL.

Henderson, P. & Morris, J. H. (1976) *A Lazy Evaluator*. In *Third Annual ACM Symposium on Principles of Programming Languages, Atlanta*, pp. 95-193, January.

Henderson, P. (1980) *Functional Programming*, Prentice-Hall.

Hill, M. D. (1984) Experimental evaluation of on-chip microprocessor cache memories, *Proc. Eleventh Int. Symp. on Computer Architecture*, Ann Arbor, June.

Hudak, P. & Goldberg, B. (1985) Serial combinators. In *Conference on Functional Programming Languages and Computer Architecture, Nancy*, Springer Verlag, pp. 382-399.

Hudak, P., Wadler, P., et al. (1988) *Report on the Functional Programming Language Haskell — Draft Proposed Standard*, Research Report YALEU/DCS/RR-666, December.

Hughes, R. J. (1982) Supercombinators: a new implementation method for applicative languages. In *Proceedings of the 1982 ACM Symposium on Lisp and Functional Programming, Pittsburgh*, ACM, August.

Hughes, R. J. (1984) *Why Functional Programming Matters*, Chalmers University of Technology, Goteborg Sweden.

Johnsson, T. (1984) Efficient compilation of lazy evaluation. In *Proceedings of the ACM Conference on Compiler Construction, Montreal*, pp. 58-69, June.

Kabakibo, A., Milutinovic, V., Silbey, A. & Furht, B. (1987) A survey of cache memory in modern microcomputer and minicomputer systems. In: Gajski, D., Milutinovic, V., Siegel, H. & Furht, B. (eds.) *Tutorial: Computer Architecture*, IEEE Computer Society Press, pp. 210-227.

Kogge, P. (1982) An architectural trail to threaded-code systems. *Computer,* **15**(3):22-32, March.

Koopman, P. (1989) *Stack Computers*, Ellis Horwood, Chichester.

Kranz, D., Kelsey, R., Rees, J., Hudak, P., Philbin, J. & Adams, N. (1986) ORBIT: an optimizing compiler for Scheme. In *Proceedings of the SIGPLAN '86 Symposium on Compiler Construction, Palo Alto, June,* ACM Press, pp. 219-233.

Landin, P. J. (1966) The next 700 programming languages. *Communications of the ACM*, **9**(3):157-64.

Moore, C. (1980) The evolution of Forth, an unusual language. *Byte,*, **5**(8):76-92, August.

Norman, A. C. (1988) Faster combinator reduction using stock hardware. In *Proceedings of the 1988 ACM Conference on Lisp and Functional Programming, Snowbird Utah*, ACM, pp. 235-243, July.

Peyton Jones, S. L. (1987) *The Implementation of Functional Programming Languages*, Prentice-Hall, London.

Peyton Jones, S. L., Clack, C., Salkild, J. & Hardie, M. (1987) GRIP — a high-performance architecture for parallel graph reduction. In Kahn, G. (ed.), *Proceedings of the Conference on Functional Programming and Computer Architecture, Portland*, Springer Verlag, pp. 98-110.

Peyton Jones, S. L. & Salkild, J. (1989) The spineless tagless G-machine. In *The Fourth International Conference on Functional Programming Languages and Computer Architecture, London*, pp. 184-201, September.

Pountaine, D. (1986) The Acorn RISC machine, *Byte*, **11**(1):387-393 January.

Przybylski, S., Horowitz, M. & Hennessy, J. (1988) Performance tradeoffs in cache design. In *The 15th Annual International Symposium on Computer Architecture, Honolulu, Hawaii, 30 May - 2 June*, IEEE Computer Society Press, pp. 290-298.

Reynolds, J. C. (1972) *Notes on a lattice-theoretic approach to the theory of computation.* Dept. Syst. and Inform. Sci., Syracuse U., Syracuse, New York.

Scheevel, M. (1986) NORMA: A graph reduction processor. In *Proceedings of the 1986 ACM Conference on Lisp and Functional Programming, Cambridge Massachusetts*, ACM, pp. 212-219, August.

Siewiorek, D. & Koopman, P. (1989) *A Case Study of a Parallel, Vector Workstation: the Titan Architecture*, Academic Press, Boston. In Press.

Smith, A. J. (1982) Cache memories, *ACM Computing Surveys*, **14**(3):473-530, September.

Stallman, R. (1988) GNU Project C Compiler. In *UNIX Programmer's Manual*, on-line system documentation, Unix version 4.3.

Turner, D. A. (1976) *SASL Reference Manual*, University of St. Andrews.

Turner, D. A. (1979a) A new implementation technique for applicative languages. *Software - Practice and Experience*, **9**(1):31-49, January.

Turner, D. A. (1979b) Another algorithm for bracket abstraction. *Journal of Symbolic Logic*, **44**(2):67-270.

Turner, D. A. (1985) Miranda -- a non-strict functional language with polymorphic types. In *Conference on Functional Programming Languages and Computer Architecture, Nancy*, Springer Verlag, pp. 1-16.

van der Hoeven, G. (1985) Ad Hoc Combinators. University of Twente, the Netherlands.

Vuilleman, J. E. (1973) *Proof Techniques for recursive programs*. Technical Report STAN-CS-73-393, Computer Science Department, Stanford University.

Wray, S. C. & Fairbairn, J. (1988) Non-strict languages - programming and implementation (draft), October 16.

Wray, S. C. (1988) Private communication, October 24.