

References

- [Aamodt and Chow, 2000] Aamodt, T. and Chow, P. (2000). Embedded ISA support for enhanced floating-point to fixed-point ANSI C compilation. *3rd ACM Intern. Conf. on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*, pages 128–137.
- [Absint, 2002] Absint (2002). Absint: WCET analyses. <http://www.absint.de/wcet.htm>.
- [Accellera, 2002] Accellera (2002). EDA industry working groups (for Rosetta). <http://www.eda.org>.
- [Accellera, 2005] Accellera (2005). SystemVerilog. <http://www.systemverilog.org>.
- [Ambler, 2005] Ambler, S. (2005). The diagrams of UML 2.0. <http://www.agilemodeling.com/essays/umlDiagrams.htm>.
- [Araujo and Malik, 1995] Araujo, G. and Malik, S. (1995). Optimal code generation for embedded memory non-homogenous register architectures. *8th Int. Symp. on System Synthesis (ISSS)*, pages 36–41.
- [August et al., 1997] August, D. I., Hwu, W. W., and Mahlke, S. (1997). A framework for balancing control flow and predication. *Ann. Workshop on Microprogramming and Microarchitecture (MICRO)*, pages 92–103.
- [Awad et al., 1996] Awad, M., Kuusela, J., and Ziegler, J. (1996). *Object-Oriented Technology for Real-Time Systems*. Prentice Hall.
- [Azevedo et al., 2002] Azevedo, A., Issenin, I., Cornea, R., Gupta, R., Dutt, N., Veidenbaum, A., and Nicolau, A. (2002). Profile-based dynamic voltage scheduling using program checkpoints. *Design, Automation, and Test in Europe (DATE)*, pages 168–175.
- [Balarin et al., 1998] Balarin, F., Lavagno, L., Murthy, P., and Sangiovanni-Vincentelli, A. (1998). Scheduling for embedded real-time systems. *IEEE Design & Test of Computers*, pages 71–82.
- [Ball, 1996] Ball, S. R. (1996). *Embedded Microprocessor Systems - Real world designs*. Newnes.
- [Ball, 1998] Ball, S. R. (1998). *Debugging Embedded Microprocessor Systems*. Newnes.

- [Barr, 1999] Barr, M. (1999). *Programming Embedded Systems*. O'Reilly.
- [Basu et al., 1999] Basu, A., Leupers, R., and Marwedel, P. (1999). Array index allocation under register constraints. *Int. Conf. on VLSI Design, Goa/India*.
- [Benini and Micheli, 1998] Benini, L. and Micheli, G. D. (1998). *Dynamic Power Management – Design Techniques and CAD Tools*. Kluwer Academic Publishers.
- [Berger, 2001] Berger, H. (2001). *Automating with STEP 7 in LAD and FBD: SIMATIC S7-300/400 Programmable Controllers*. Wiley.
- [Bergé et al., 1995] Bergé, J.-M., Levia, O., and Rouillard, J. (1995). *High-Level System Modeling*. Kluwer Academic Publishers.
- [Berkelaar and et al., 2005] Berkelaar, M. and et al. (2005). Unix[™] manual page of lp_solve. Eindhoven University of Technology, Design Automation Section, current version of source code available at http://groups.yahoo.com/group/lp_solve/files.
- [Bieker and Marwedel, 1995] Bieker, U. and Marwedel, P. (1995). Retargetable self-test program generation using constraint logic programming. *32nd Design Automation Conference (DAC)*, pages 605–611.
- [Boussinot and de Simone, 1991] Boussinot, F. and de Simone, R. (1991). The Esterel language. *Proc. of the IEEE, Vol. 79, No. 9*, pages 1293–1304.
- [Bouyssounouse and Sifakis, 2005] Bouyssounouse, B. and Sifakis, J., editors (2005). *Embedded Systems Design, The ARTIST Roadmap for Research and Development*. Lecture Notes in Computer Science, Vol. 3436, Springer.
- [Brahme and Abraham, 1984] Brahme, D. and Abraham, J. A. (1984). Functional testing of microprocessors. *IEEE Trans. on Computers*, pages 475–485.
- [Bremaud, 1999] Bremaud, P. (1999). *Markov Chains*. Springer Verlag.
- [Burd and Brodersen, 2000] Burd, T. and Brodersen, R. (2000). Design issues for dynamic voltage scaling. *Intern. Symp. on Low Power Electronics and Design (ISLPED)*, pages 9–14.
- [Burd and Brodersen, 2003] Burd, T. and Brodersen, R. W. (2003). *Energy efficient microprocessor design*. Kluwer Academic Publishers.
- [Burkhardt, 2001] Burkhardt, J. (2001). *Pervasive Computing*. Addison-Wesley.
- [Burns and Wellings, 1990] Burns, A. and Wellings, A. (1990). *Real-Time Systems and Their Programming Languages*. Addison-Wesley.
- [Burns and Wellings, 2001] Burns, A. and Wellings, A. (2001). *Real-Time Systems and Programming Languages (Third Edition)*. Addison Wesley.
- [Buttazzo, 2002] Buttazzo, G. (2002). *Hard Real-time computing systems*. Kluwer Academic Publishers, 4th printing.
- [Byteflight Consortium, 2003] Byteflight Consortium (2003). Home page. <http://www.byteflight.com>.
- [Camposano and Wolf, 1996] Camposano, R. and Wolf, W. (1996). Message from the editors-in-chief. *Design Automation for Embedded Systems*.

- [Center for Embedded Computer Systems, 2003] Center for Embedded Computer Systems (2003). SoC Environment. <http://www.cecs.uci.edu/~cad/sce.html>.
- [Chandrakasan et al., 1992] Chandrakasan, A., Sheng, S., and Brodersen, R. W. (1992). Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4):119–123.
- [Chandrakasan et al., 1995] Chandrakasan, A. P., Sheng, S., and Brodersen, R. W. (1995). Low power CMOS digital design. *Kluwer Academic Publishers*.
- [Chetto et al., 1990] Chetto, H., Silly, M., and Bouchentouf, T. (1990). Dynamic scheduling of real-time tasks under precedence constraints. *Journal of Real-Time Systems*, 2.
- [Chung et al., 2001] Chung, E.-Y., Benini, L., and Micheli, G. D. (2001). Source code transformation based on software cost analysis. In *Int. Symp. on System Synthesis (ISSS)*, pages 153–158.
- [Cinderella ApS, 2003] Cinderella ApS (2003). home page. <http://www.cinderella.dk>.
- [Clarke and et al., 2003] Clarke, E. and et al. (2003). Model checking@CMU. <http://www-2.cs.cmu.edu/~modelcheck/index.html>.
- [Clarke et al., 2000] Clarke, E. M., Grumberg, O., and Peled, D. A. (2000). *Model Checking*. The MIT Press; Second printing.
- [Clouard et al., 2003] Clouard, A., Jain, K., Ghenassia, F., Maillet-Contoz, L., and Strassen, J. (2003). Using transactional models in SoC design flow. in [Müller et al., 2003], pages 29–64.
- [Coelho, 1989] Coelho, D. R. (1989). The VHDL handbook. *Kluwer Academic Publishers*.
- [Cortadella et al., 2000] Cortadella, J., Kondratyev, A., Lavagno, L., Massot, M., Moral, S., Passerone, C., Watanabe, Y., and Sangiovanni-Vincentelli, A. (2000). Task generation and compile-time scheduling for mixed data-control embedded software. *Design automation conference (DAC)*, pages 489–494.
- [Dasgupta, 1979] Dasgupta, S. (1979). The organization of microprogram stores. *ACM Computing Surveys*, Vol. 11, pages 39–65.
- [Davis et al., 2001] Davis, J., Hylands, C., Janneck, J., Lee, E. A., Liu, J., Liu, X., Neuendorfer, S., Sachs, S., Stewart, M., Vissers, K., Whitaker, P., and Xiong, Y. (2001). Overview of the Ptolemy project. *Technical Memorandum UCB/ERL M01/11*; <http://ptolemy.eecs.berkeley.edu>.
- [De Greef et al., 1997a] De Greef, E., Catthoor, F., and Man, H. D. (1997a). Array placement for storage size reduction in embedded multimedia systems. *IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*, pages 66–75.
- [De Greef et al., 1997b] De Greef, E., F.Catthoor, and Man, H. (1997b). Memory size reduction through storage order optimization for embedded parallel multimedia applications. *Intern. Parallel Proc. Symp.(IPPS) in Proc. Workshop on “Parallel Processing and Multimedia”*, pages 84–98.
- [De Man, 2002] De Man, H. (2002). Keynote session at DATE’02. <http://www.date-conference.com/conference/2002/keynotes/index.htm>.

- [Deutsches Institut für Normung, 1997] Deutsches Institut für Normung (1997). *DIN 66253, Programmiersprache PEARL, Teil 2 PEARL 90*. Beuth-Verlag; English version available through <http://www.din.de>.
- [Dierickx, 2000] Dierickx, B. (2000). CMOS image sensors - concepts, Photonics West 2000 short course. <http://www.fillfactory.com/html/technology/html/publications.htm>.
- [Douglass, 2000] Douglass, B. P. (2000). *Real-Time UML, 2nd edition*. Addison Wesley.
- [Drusinsky and Harel, 1989] Drusinsky, D. and Harel, D. (1989). Using statecharts for hardware description and synthesis. *IEEE Trans. on Computer Design*, pages 798–807.
- [Dunn, 2002] Dunn, W. (2002). *Practical Design of Safety-Critical Computer Systems*. Reliability Press.
- [Elsevier B.V., 2003a] Elsevier B.V. (2003a). Sensors and actuators A: Physical. *An International Journal*.
- [Elsevier B.V., 2003b] Elsevier B.V. (2003b). Sensors and actuators B: Chemical. *An International Journal*.
- [Esterel, 2002] Esterel, T. I. (2002). Homepage. <http://www.esterel-technologies.com>.
- [Falk and Marwedel, 2003] Falk, H. and Marwedel, P. (2003). Control flow driven splitting of loop nests at the source code level. *Design, Automation and Test in Europe (DATE)*, pages 410–415.
- [Fettweis et al., 1998] Fettweis, G., Weiss, M., Drescher, W., Walther, U., Engel, F., Kobayashi, S., and Richter, T. (1998). Breaking new grounds over 3000 MMAC/s: a broadband mobile multimedia modem DSP. *Intern. Conf. on Signal Processing Application & Technology (ICSPA)*, available at <http://citeseer.ist.psu.edu/111037.html>.
- [Fisher and Dietz, 1998] Fisher, R. and Dietz, H. G. (1998). Compiling for SIMD within a single register. *Annual Workshop on Lang. & Compilers for Parallel Computing (LCPC)*, pages 290–304.
- [Fisher and Dietz, 1999] Fisher, R. J. and Dietz, H. G. (1999). The Scc compiler: SWARing at MMX and 3DNow! *Annual Workshop on Lang. & Compilers for Parallel Computing (LCPC)*, pages 399–414.
- [FlexRay Consortium, 2002] FlexRay Consortium (2002). Flexray[®] requirement specification. version 2.01. <http://www.flexray.de>.
- [Fowler and Scott, 1998] Fowler, M. and Scott, K. (1998). *UML Distilled - Applying the Standard Object Modeling Language*. Addison-Wesley.
- [Fu et al., 1987] Fu, K., Gonzalez, R., and Lee, C. (1987). *Robotics*. McGraw-Hill.
- [Gajski et al., 1994] Gajski, D., Vahid, F., Narayan, S., and Gong, J. (1994). *Specification and Design of Embedded Systems*. Prentice Hall.
- [Gajski et al., 2000] Gajski, D., Zhu, J., Dömer, R., Gerstlauer, A., and Zhao, S. (2000). *SpecC: Specification Language Methodology*. Kluwer Academic Publishers.

- [Ganssle, 1992] Ganssle, J. G. (1992). *Programming Embedded Systems*. Academic Press.
- [Ganssle, 2000] Ganssle, J. G. (2000). *The Art of Designing Embedded Systems*. Newnes.
- [Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability*. Bell Laboratories, Murray Hill, New Jersey.
- [Geffroy and Motet, 2002] Geffroy, J.-C. and Motet, G. (2002). *Design of Dependable computing Systems*. Kluwer Academic Publishers.
- [Gelsen, 2003] Gelsen, O. (2003). Organic displays enter consumer electronics. *Opto & Laser Europe, June*; available at <http://optics.org/articles/ole/8/6/6/1>.
- [Gerstlauer et al., 2001] Gerstlauer, A., Dömer, R., Peng, J., and Gajski, D. (2001). *System Design: A Practical Guide with SystemC*. Kluwer Academic Publishers.
- [Ghez et al., 2000] Ghez, C., Miranda, M., Vandecapelle, A., et al. (2000). Systematic high-level address code transformations for piece-wise linear indexing. *Proc. of SIPS*.
- [Gupta, 1998] Gupta, R. (1998). Introduction to embedded systems. <http://www.ics.uci.edu/~rgupta/ics212.html>.
- [Halbwachs, 1998] Halbwachs, N. (1998). Synchronous programming of reactive systems, a tutorial and commented bibliography. *Tenth International Conference on Computer-Aided Verification, CAV'98, LNCS 1427, Springer Verlag*; see also: <http://www-verimag.imag.fr/PEOPLE/Nicolas.Halbwachs/cav98tutorial.html>.
- [Halbwachs et al., 1991] Halbwachs, N., Caspi, P., Raymond, P., and Pilaud, D. (1991). The synchronous dataflow language LUSTRE. *Proc. of the IEEE*, 79:1305–1320.
- [Hansmann, 2001] Hansmann, U. (2001). *Pervasive Computing*. Springer Verlag.
- [Harbour, 1993] Harbour, M. G. (1993). RT-POSIX: An overview. <http://www.ctr.unican.es/publications/mgh-1993a.pdf>.
- [Harel, 1987] Harel, D. (1987). StateCharts: A visual formalism for complex systems. *Science of Computer Programming*, pages 231–274.
- [Hayes, 1982] Hayes, J. (1982). A unified switching theory with applications to VLSI design. *Proceedings of the IEEE, Vol.70*, pages 1140–1151.
- [Healy et al., 1999] Healy, C., Arnold, R., Mueller, F., Whalley, D., and Harmon, M. (1999). Bounding pipeline and instructions cache performance. *IEEE Transactions on Computers*, pages 53–70.
- [Hennessy and Patterson, 1995] Hennessy, J. L. and Patterson, D. A. (1995). *Computer Organization – The Hardware/Software Interface*. Morgan Kaufmann Publishers Inc.
- [Hennessy and Patterson, 1996] Hennessy, J. L. and Patterson, D. A. (1996). *Computer Architecture – A Quantitative Approach*. Morgan Kaufmann Publishers Inc.
- [Herrera et al., 2003a] Herrera, F., Fernández, V., Sánchez, P., and Villar, E. (2003a). Embedded software generation from SystemC for platform based design. in [Müller et al., 2003], pages 247–272.

- [Herrera et al., 2003b] Herrera, F., Posadas, H., Sánchez, P., and Villar, E. (2003b). Systemic embedded software generation from SystemC. *Design, Automation and Test in Europe (DATE)*, pages 10142–10149.
- [Hoare, 1985] Hoare, C. (1985). *Communicating Sequential Processes*. Prentice Hall International Series in Computer Science.
- [Horn, 1974] Horn, W. (1974). Some simple scheduling algorithms. *Naval Research Logistics Quarterly*, Vol. 21, pages 177–185.
- [Huerlimann, 2003] Huerlimann, D. (2003). Opentrack home page. <http://www.opentrack.ch>.
- [Hüls, 2002] Hüls, T. (2002). Optimizing the energy consumption of an MPEG application (in German). *Master thesis, CS Dept., Univ. Dortmund*, <http://ls12-www.cs.uni-dortmund.de/publications/theses>.
- [IBM Inc., 2002] IBM Inc. (2002). Security: User authentication. <http://www.pc.ibm.com/us/security/userauth.html>.
- [IEC, 2002] IEC (2002). GRAFCET specification language for sequential function charts. <http://tc3.iec.ch/txt/147.htm>.
- [IEEE, 1997] IEEE (1997). *IEEE Standard VHDL Language Reference Manual (1076-1997)*. IEEE.
- [IEEE, 1992] IEEE, D. (1992). Draft standard VHDL language reference manual. *IEEE Standards Department, 1992*.
- [IMEC, 1997] IMEC (1997). LIC-SMARTpen identifies signer. *IMEC Newsletter*, http://www.imec.be/wwwinter/mediacenter/en/newsletter_18.pdf.
- [IMEC, 2003] IMEC (2003). Design technology program. http://www.imec.be/ovinter/static_research/designtechnology.shtml.
- [IMEC Desics group, 2003] IMEC Desics group (2003). Task concurrency management (overview of IMEC activities). <http://www.imec.be/design/tcm/>.
- [Intel, 2005] Intel (2005). Intel Itanium 2 processor. <http://www.intel.com/products/processor/itanium2>.
- [Ishihara and Yasuura, 1998] Ishihara, T. and Yasuura, H. (1998). Voltage scheduling problem for dynamically variable voltage processors. *Intern. Symp. on Low Power Electronics and Design (ISLPED)*, pages 197–2002.
- [Iyer and Marculescu, 2002] Iyer, A. and Marculescu, D. (2002). Power and performance evaluation of globally asynchronous locally synchronous processors. *Intern. Symp. on Computer Architecture (ISCA)*, pages 158–168.
- [Jackson, 1955] Jackson, J. (1955). Scheduling a production line to minimize maximum tardiness. *Management Science Research Project 43, University of California, Los Angeles*.
- [Jacome et al., 2000] Jacome, M., de Veciana, G., and Lapinskii, V. (2000). Exploring performance tradeoffs for clustered VLIW ASIPs. *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pages 504–510.

- [Jacome and de Veciana, 1999] Jacome, M. F. and de Veciana, G. (1999). Lower bound on latency for VLIW ASIP datapaths. *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pages 261–269.
- [Jain et al., 2001] Jain, M., Balakrishnan, M., and Kumar, A. (2001). ASIP design methodologies : Survey and issues. *Fourteenth International Conference on VLSI Design*, pages 76–81.
- [Janka, 2002] Janka, R. (2002). *Specification and Design Methodology for Real-Time Embedded Systems*. Kluwer Academic Publishers.
- [Jantsch, 2003] Jantsch, A. (2003). *Modeling Embedded Systems and SoC's: Concurrency and Time in Models of Computation*. Morgan Kaufmann.
- [Java Community Process, 2002] Java Community Process (2002). JSR-1 – real-time specification for Java. <http://www.jcp.org/en/jsr/detail?id=1>.
- [Jeffrey and Leduc, 1996] Jeffrey, A. and Leduc, G. (1996). E-LOTOS core language. <http://citeseer.ist.psu.edu/jeffrey96elotos.html>.
- [Jha and Dutt, 1993] Jha, P. and Dutt, N. (1993). Rapid estimation for parameterized components in high-level synthesis. *IEEE Transactions on VLSI Systems*, pages 296–303.
- [Jones, 1997] Jones, M. (1997). What really happened on Mars Rover Pathfinder. in: *P.G. Neumann (ed.): comp.risks, The Risks Digest, Vol. 19, Issue 49*; available at <http://www.cs.berkeley.edu/~brewer/cs262/PriorityInversion.html>.
- [Kahn, 1974] Kahn, G. (1974). The semantics of a simple language for parallel programming. *Proc. of the IFIP Congress 74*, pages 471–475.
- [Keding et al., 1998] Keding, H., Willems, M., Coors, M., and Meyr, H. (1998). FRIDGE: A fixed-point design and simulation environment. *Design, Automation and Test in Europe (DATE)*, pages 429–435.
- [Kempe, 1995] Kempe, M. (1995). Ada 95 reference manual, iso/iec standard 8652:1995. (*HTML-version*), <http://www.adahome.com/rm95/>.
- [Kernighan and Ritchie, 1988] Kernighan, B. W. and Ritchie, D. M. (1988). *The C Programming Language*. Prentice Hall.
- [Kienhuis et al., 2000] Kienhuis, B., Rijjkema, E., and Deprettere, E. (2000). Compaan: Deriving process networks from Matlab for embedded signal processing architectures. *Proc. 8th Intern. Workshop on Hardware/Software Codesign (CODES)*.
- [Kobryn, 2001] Kobryn, C. (2001). UML 2001: A standardization Odyssey. *Communication of the ACM (CACM)*, available at http://www.omg.org/attachments/pdf/UML_2001_CACM_Oct99_p29-Kobryn.pdf, pages 29–36.
- [Kohavi, 1987] Kohavi, Z. (1987). *Switching and Finite Automata Theory*. Tata McGraw-Hill Publishing Company, New Delhi, 9th reprint.
- [Koninklijke Philips Electronics N.V., 2003] Koninklijke Philips Electronics N.V. (2003). Ambient intelligence. <http://www.philips.com/research/ami>.

- [Kopetz, 1997] Kopetz, H. (1997). *Real-Time Systems –Design Principles for Distributed Embedded Applications–*. Kluwer Academic Publishers.
- [Kopetz, 2003] Kopetz, H. (2003). Architecture of safety-critical distributed real-time systems. *Invited Talk; Design, Automation, and Test in Europe (DATE)*.
- [Kopetz and Grunsteidl, 1994] Kopetz, H. and Grunsteidl, G. (1994). TTP –a protocol for fault-tolerant real-time systems. *IEEE Computer*, 27:14–23.
- [Krall, 2000] Krall, A. (2000). Compilation techniques for multimedia extensions. *International Journal of Parallel Programming*, 28:347–361.
- [Krishna and Shin, 1997] Krishna, C. and Shin, K. G. (1997). *Real-Time Systems*. McGraw-Hill, Computer Science Series.
- [Kropf, 1999] Kropf, T. (1999). *Introduction to Formal Hardware Verification*. Springer-Verlag.
- [Krstic and Dey, 2002] Krstic, A. and Dey, S. (2002). Embedded software-based self-test for programmable core-based designs. *IEEE Design & Test*, pages 18–27.
- [Krüger, 1986] Krüger, G. (1986). Automatic generation of self-test programs: A new feature of the MIMOLA design system. *23rd Design Automation Conference (DAC)*, pages 378–384.
- [Kuchcinski, 2002] Kuchcinski, K. (2002). System partitioning (course notes). http://www.cs.lth.se/home/Krzysztof_Kuchcinski/DES/Lectures/Lecture7.pdf.
- [Kwok and Ahmad, 1999] Kwok, Y.-K. and Ahmad, I. (1999). Static scheduling algorithms for allocation directed task graphs to multiprocessors. *ACM Computing Surveys*, 31:406–471.
- [Könemann et al., 1979] Könemann, B., Mucha, J., and Zwiehoff, G. (1979). Built-in logic block observer. *Proc. IEEE Intern. Test Conf.*, pages 261–266.
- [Lam et al., 1991] Lam, M. S., Rothberg, E. E., and Wolf, M. E. (1991). The cache performance and optimizations of blocked algorithms. *Proceedings of ASPLOS IV*, pages 63–74.
- [Landwehr and Marwedel, 1997] Landwehr, B. and Marwedel, P. (1997). A new optimization technique for improving resource exploitation and critical path minimization. *10th International Symposium on System Synthesis (ISSS)*, pages 65–72.
- [Lapinskii et al., 2001] Lapinskii, V., Jacome, M. F., and de Veciana, G. (2001). Application-specific clustered VLIW datapaths: Early exploration on a parameterized design space. Technical Report UT-CERC-TR-MFJ/GDV-01-1, Computer Engineering Research Center, University of Texas at Austin.
- [Laprie, 1992] Laprie, J. C., editor (1992). *Dependability: basic concepts and terminology in English, French, German, Italian and Japanese*. IFIP WG 10.4, Dependable Computing and Fault Tolerance, in: volume 5 of Dependable computing and fault tolerant systems, Springer Verlag.
- [Larsen and Amarasinghe, 2000] Larsen, S. and Amarasinghe, S. (2000). Exploiting superword parallelism with multimedia instructions sets. *Programming Language Design and Implementation (PLDI)*, pages 145–156.

- [Lawler, 1973] Lawler, E. L. (1973). Optimal sequencing of a single machine subject to precedence constraints. *Managements Science*, Vol. 19, pages 544–546.
- [Lee, 1999] Lee, E. (1999). Embedded software – an agenda for research. Technical report, UCB ERL Memorandum M99/63.
- [Lee and Messerschmitt, 1987] Lee, E. and Messerschmitt, D. (1987). Synchronous data flow. *Proc. of the IEEE*, vol. 75, pages 1235–1245.
- [Lee et al., 2001] Lee, S., Ermedahl, A., , and Min, S. (2001). An accurate instruction-level energy consumption model for embedded risc processors. *ACM SIGPLAN Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*.
- [Leupers, 1997] Leupers, R. (1997). *Retargetable Code Generation for Digital Signal Processors*. Kluwer Academic Publishers.
- [Leupers, 1999] Leupers, R. (1999). Exploiting conditional instructions in code generation for embedded VLIW processors. *Design, Automation and Test in Europe (DATE)*.
- [Leupers, 2000a] Leupers, R. (2000a). *Code Optimization Techniques for Embedded Processors - Methods, Algorithms, and Tools*. Kluwer Academic Publishers.
- [Leupers, 2000b] Leupers, R. (2000b). Code selection for media processors with SIMD instructions. *Design, Automation and Test in Europe (DATE)*, pages 4–8.
- [Leupers, 2000c] Leupers, R. (2000c). Instruction scheduling for clustered VLIW DSPs. *Int. Conf. on Parallel Architectures and Compilation Techniques (PACT)*, pages 291–300.
- [Leupers and David, 1998] Leupers, R. and David, F. (1998). A uniform optimization technique for offset assignment problems. *Int. Symp. on System Synthesis (ISSS)*, pages 3–8.
- [Leupers and Marwedel, 1995] Leupers, R. and Marwedel, P. (1995). Time-constrained code compaction for DSPs. *Int. Symp. on System Synthesis (ISSS)*, pages 54–59.
- [Leupers and Marwedel, 1996] Leupers, R. and Marwedel, P. (1996). Algorithms for address assignment in DSP code generation. *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pages 109–112.
- [Leupers and Marwedel, 1999] Leupers, R. and Marwedel, P. (1999). Function inlining under code size constraints for embedded processors. *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pages 253–256.
- [Leupers and Marwedel, 2001] Leupers, R. and Marwedel, P. (2001). *Retargetable Compiler Technology for Embedded Systems – Tools and Applications*. Kluwer Academic Publishers.
- [Leveson, 1995] Leveson, N. (1995). *Safeware, System safety and Computers*. Addison Wesley.
- [Liao et al., 1995a] Liao, S., Devadas, S., Keutzer, K., and Tijang, S. (1995a). Code optimization techniques for embedded DSP microprocessors. *32nd Design Automation Conference (DAC)*, pages 599–604.
- [Liao et al., 1995b] Liao, S., Devadas, S., Keutzer, K., Tijang, S., and Wang, A. (1995b). Storage assignment to decrease code size. *Programming Language Design and Implementation (PLDI)*, pages 186–195.

- [Liu and Layland, 1973] Liu, C. L. and Layland, J. W. (1973). Scheduling algorithms for multi-programming in a hard real-time environment. *Journal of the Association for Computing Machinery (JACM)*, pages 40–61.
- [Liu, 2000] Liu, J. W. (2000). *Real-Time Systems*. Prentice Hall.
- [Lorenz et al., 2002] Lorenz, M., Wehmeyer, L., Draeger, T., and Leupers, R. (2002). Energy aware compilation for DSPs with SIMD instructions. *LCTES/SCOPES '02*, pages 94–101.
- [Machanik, 2002] Machanik, P. (2002). Approaches to addressing the memory wall. *Technical Report, November, Univ. Brisbane*.
- [Mahlke et al., 1992] Mahlke, S. A., Lin, D. C., Chen, W. Y., Hank, R. E., and Bringmann, R. A. (1992). Effective compiler support for predicated execution using the hyperblock. *MICRO-92*, pages 45–54.
- [Marwedel and Goossens, 1995] Marwedel, P. and Goossens, G., editors (1995). *Code Generation for Embedded Processors*. Kluwer Academic Publishers.
- [Marwedel and Schenk, 1993] Marwedel, P. and Schenk, W. (1993). Cooperation of synthesis, retargetable code generation and testgeneration in the MSS. *EDAC-EUROASIC'93*, pages 63–69.
- [Marzano and Aarts, 2003] Marzano, S. and Aarts, E. (2003). *The New Everyday*. 010 Publishers.
- [McLaughlin and Moore, 1998] McLaughlin, M. and Moore, A. (1998). Real-Time Extensions to UML. <http://www.ddj.com/articles/1998/9812g/9812.htm>.
- [Menard and Sentieys, 2002] Menard, D. and Sentieys, O. (2002). Automatic evaluation of the accuracy of fixed-point algorithms. *Design, Automation and Test in Europe (DATE)*, pages 529–535.
- [Mermet et al., 1998] Mermet, J., Marwedel, P., Ramming, F. J., Newton, C., Borrione, D., and Lefaou, C. (1998). Three decades of hardware description languages in europe. *Journal of Electrical Engineering and Information Science*, 3:106pp.
- [Microsoft Inc., 2003] Microsoft Inc. (2003). Microsoft windows embedded. microsoft.com/windows/embedded/default.aspx.
- [Miranda et al., 2004] Miranda, M., Brockmeyer, E., Meeuwen, T. V., Ghez, C., and Catthoor, F. (2004). Low power data transfer and communication for SoC. in: *C. Piquet (ed.): Lower Power Electronics and Design, CRC Press, (to be published)*.
- [Miranda et al., 1998] Miranda, M., F.Catthoor, M.Janssen, and Man, H. (1998). High-level address optimisation and synthesis techniques for data-transfer intensive applications. *IEEE Trans. on VLSI Systems*, pages 677–686.
- [MPI/RT forum, 2001] MPI/RT forum (2001). Document for the real-time message passing interface (mpi/rt-1.1. <http://www.mpirt.org/drafts/mpirt-report-18dec01.pdf>.
- [Muchnick, 1997] Muchnick, S. S. (1997). *Advanced compiler design and implementation*. Morgan Kaufmann Publishers, Inc.

- [Müller et al., 2003] Müller, W., Rosenstiel, W., and Ruf, J. (2003). *SystemC – Methodologies and Applications*. Kluwer Academic Publications.
- [Neumann, 1995] Neumann, P. G. (1995). *Computer Related Risks*. Addison Wesley.
- [Niemann, 1998] Niemann, R. (1998). *Hardware/Software Co-Design for Data-Flow Dominated Embedded Systems*. Kluwer Academic Publishers.
- [Nilsen, 2004] Nilsen, K. (2004). Real-Time Java. <http://www.newmonics.com/about/tech/realjava.shtml>.
- [Object Management Group (OMG), 2002] Object Management Group (OMG) (2002). Real-time CORBA specification, version 1.1, august 2002. *Object Management Group*, <http://www.omg.org/docs/formal/02-08-02.ps>.
- [Object Management Group (OMG), 2003] Object Management Group (OMG) (2003). CORBA@basics. <http://www.omg.org/gettingstarted/corbafaq.htm>.
- [OMG, 2005] OMG (2005). UMLTM resource page. <http://www.uml.org>.
- [Oppenheim et al., 1999] Oppenheim, A. V., Schafer, R., and Buck, J. R. (1999). *Digital Signal Processing*. Pearson Higher Education.
- [Österling et al., 1997] Österling, A., Benner, T., Ernst, R., Herrmann, D., Scholz, T., and Ye, W. (1997). The COSYMA system. <http://www.ida.ing.tu-bs.de/research/projects/cosymal/overview/node1.html>.
- [Palkovic et al., 2002] Palkovic, M., Miranda, M., and Catthoor, F. (2002). Systematic power-performance trade-off in MPEG-4 by means of selective function inlining steered by address optimisation opportunities. *Design, Automation, and Test in Europe (DATE)*, pages 1072–1079.
- [palowireless, 2003] palowireless (2003). HomeRF Resource Center. <http://www.homerf.org>.
- [Parker, 1992] Parker, K. P. (1992). *The Boundary Scan Handbook*. Kluwer Academic Press.
- [Pino and Lee, 1995] Pino, J. L. and Lee, E. (1995). Hierarchical static scheduling of dataflow graphs onto multiple processors. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*.
- [Poseidon, 2003] Poseidon (2003). Documentation for Poseidon for UML. <http://www.gentleware.com/index.php?id=documentation>.
- [Press, 2003] Press, D. (2003). *Guidelines for Failure Mode and Effects Analysis for Automotive, Aerospace and General Manufacturing Industries*. CRC Press.
- [Ramamritham, 2002] Ramamritham, K. (2002). System support for real-time embedded systems. *in: Tutorial 1, 39th Design Automation Conference (DAC)*.
- [Ramamritham et al., 1998] Ramamritham, K., Shen, C., Gonzalez, O., Sen, S., and Shirgurkar, S. B. (1998). Using Windows NT for real-time applications: Experimental observations and recommendations. *IEEE Real-Time Technology and Applications Symposium (RTAS)*, pages 102–111.
- [Reisig, 1985] Reisig, W. (1985). *Petri nets*. Springer Verlag.

- [Rixner et al., 2000] Rixner, S., Dally, W. J., Khailany, B., Mattson, P., Kapasi, U. J., and Owens, J. D. (2000). Register organization for media processing. *6th Intern. Symp. on High-Performance Computer Architecture*, pages 375–386.
- [Russell and Jacome, 1998] Russell, T. and Jacome, M. F. (1998). Software power estimation and optimization for high performance, 32-bit embedded processors. *Proc. International Conference on Computer Design (ICCD)*, pages 328–333.
- [Ryan, 1995] Ryan, M. (1995). Market focus – insight into markets that are making the news in EE Times. [http://techweb.cmp.com/techweb/eet/embedded/embedded.html\(Sept.11\)](http://techweb.cmp.com/techweb/eet/embedded/embedded.html(Sept.11)).
- [Sangiovanni-Vincentelli, 2002] Sangiovanni-Vincentelli, A. (2002). The context for platform-based design. *IEEE Design & Test of Computers*, page 120.
- [Schmitz et al., 2002] Schmitz, M., Al-Hashimi, B., and Eles, P. (2002). Energy-efficient mapping and scheduling for dvs enabled distributed embedded systems. *Design, Automation and Test in Europe (DATE)*, pages 514–521.
- [SDL Forum Society, 2003a] SDL Forum Society (2003a). Home page. <http://www.sdl-forum.org>.
- [SDL Forum Society, 2003b] SDL Forum Society (2003b). List of commercial tools. <http://www.sdl-forum.org/Tools/Commercial.htm>.
- [Sedgewick, 1988] Sedgewick, R. (1988). *Algorithms*. Addison-Wesley.
- [SEMATECH, 2003] SEMATECH (2003). International technology roadmap for semiconductors (ITRS). <http://public.itrs.net>.
- [Sha et al., 1990] Sha, L., Rajkumar, R., and Lehoczky, J. (1990). Priority inheritance protocols: An approach to real-time synchronisation. *IEEE Trans. on Computers*, pages 1175–1185.
- [Shi and Brodersen, 2003] Shi, C. and Brodersen, R. (2003). An automated floating-point to fixed-point conversion methodology. *Int. Conf. on Audio Speed and Signal Processing (ICASSP)*, pages 529–532.
- [Simunic et al., 2000] Simunic, T., Benini, L., Acquaviva, A., Glynn, P., and Micheli, G. D. (2000). Energy efficient design of portable wireless devices. *Intern. Symp. on Low Power Electronics and Design (ISLPED)*, pages 49–54.
- [Simunic et al., 2001] Simunic, T., Benini, L., Acquaviva, A., Glynn, P., and Micheli, G. D. (2001). Dynamic voltage scaling and power management for portable systems. *Design Automation Conference (DAC)*, pages 524–529.
- [Simunic et al., 1999] Simunic, T., Benini, L., and De Micheli, G. (1999). Cycle-accurate simulation of energy consumption in embedded systems. *Design Automation Conference (DAC)*, pages 876–872.
- [Society for Display Technology, 2003] Society for Display Technology (2003). Home page. <http://www.sid.org>.
- [Spivey, 1992] Spivey, M. (1992). *The Z Notation: A Reference Manual*. Prentice Hall International Series in Computer Science, 2nd edition.

- [Stankovic and Ramamritham, 1991] Stankovic, J. and Ramamritham, K. (1991). The Spring kernel: a new paradigm for real-time systems. *IEEE Software*, 8:62–72.
- [Stankovic et al., 1998] Stankovic, J., Spuri, M., Ramamritham, K., and Buttazzo, G. (1998). *Deadline Scheduling for Real-Time Systems, EDF and related algorithms*. Kluwer Academic Publishers.
- [Steinke et al., 2002a] Steinke, S., Grunwald, N., Wehmeyer, L., Banakar, R., Balakrishnan, M., and Marwedel, P. (2002a). Reducing energy consumption by dynamic copying of instructions onto onchip memory. *Int. Symp. on System Synthesis (ISSS)*, pages 213–218.
- [Steinke et al., 2001] Steinke, S., Knauer, M., Wehmeyer, L., and Marwedel, P. (2001). An accurate and fine grain instruction-level energy model supporting software optimizations. *Proc. of the International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*.
- [Steinke et al., 2002b] Steinke, S., L. Wehmeyer, Lee, B.-S., and Marwedel, P. (2002b). Assigning program and data objects to scratchpad for energy reduction. *Design, Automation and Test in Europe (DATE)*, pages 409–417.
- [Stiller, 2000] Stiller, A. (2000). New processors (in German). *c't*, 22:52.
- [Storey, 1996] Storey, N. (1996). *Safety-critical Computer Systems*. Addison Wesley.
- [Stritter and Gunter, 1979] Stritter, E. and Gunter, T. (1979). Microprocessor architecture for a changing world: The Motorola 68000. *IEEE Computer*, 12:43–52.
- [Sudarsanam, 1997] Sudarsanam, A. (1997). SPAM compiler release. <http://www.ee.princeton.edu/~spam/release.html>.
- [Sudarsanam et al., 1997] Sudarsanam, A., Liao, S., and Devadas, S. (1997). Analysis and evaluation of address arithmetic capabilities in custom DSP architectures. *Design Automation Conference (DAC)*, pages 287–292.
- [Sudarsanam and Malik, 1995] Sudarsanam, A. and Malik, S. (1995). Memory bank and register allocation in software synthesis for ASIPs. *Intern. Conf. on Computer-Aided Design (ICCAD)*, pages 388–392.
- [Synopsys, 2005] Synopsys (2005). System studio. http://www.synopsys.com/products/cocentric_studio.
- [SystemC, 2002] SystemC (2002). Home page. <http://www.SystemC.org>.
- [Takada, 2001] Takada, H. (2001). Real-time operating system for embedded systems. in: *M. Imai and N. Yoshida (eds.): Tutorial 2 – Software Development Methods for Embedded Systems, Asia South-Pacific Design Automation Conference (ASP-DAC)*.
- [Tan et al., 2003] Tan, T. K., Raghunathan, A., and Jha, N. K. (2003). Software architectural transformations: A new approach to low energy embedded software. *Design, Automation and Test in Europe (DATE)*, pages 11046–11051.
- [Teich et al., 1999] Teich, J., Zitzler, E., and Bhattacharyya, S. (1999). 3D exploration of software schedules for DSP algorithms. *CODES'99*, page 168pp.

- [Telelogic, 1999] Telelogic (1999). Real-Time Extensions to UML. <http://www.telelogic.com/help/search/index.cfm>.
- [Telelogic AB, 2003] Telelogic AB (2003). Home page. <http://www.telelogic.com>.
- [Tensilica Inc., 2003] Tensilica Inc. (2003). Home page. <http://www.tensilica.com>.
- [Tewari, 2001] Tewari, A. (2001). *Modern Control Design with MATLAB and SIMULINK*. John Wiley and Sons Ltd.
- [The Dobelle Institute, 2003] The Dobelle Institute (2003). Home page. <http://www.dobelle.com>.
- [The SUIF group, 2003] The SUIF group (2003). SUIF compiler system. <http://suif.stanford.edu>.
- [Thiébaud, 1995] Thiébaud, D. (1995). Parallel programming in C for the transputer. <http://maven.smith.edu/~thiebaut/transputer/descript.html>.
- [Thoen and Catthoor, 2000] Thoen, F. and Catthoor, F. (2000). *Modelling, Verification and Exploration of Task-Level Concurrency in Real-Time Embedded Systems*. Kluwer Academic Publishers.
- [Thomas and Moorby, 1991] Thomas, D. E. and Moorby, P. (1991). The verilog hardware description language. *Kluwer Academic Publishers*.
- [TimeSys Inc., 2003] TimeSys Inc. (2003). Home page. <http://www.timesys.com>.
- [Tiwari et al., 1994] Tiwari, V., Malik, S., and Wolfe, A. (1994). Power analysis of embedded software: A first step towards software power minimization. *IEEE Trans. On VLSI Systems*, pages 437–445.
- [Transmeta, 2005] Transmeta, I. (2005). Support: Technical documentations. <http://www.transmeta.com/developers/techdocs.html>.
- [Vaandrager, 1998] Vaandrager, F. (1998). Lectures on embedded systems. in *Rozenberg, Vaandrager (eds), LNCS, Vol. 1494*.
- [Vahid, 1995] Vahid, F. (1995). Procedure exlining. *Int. Symp. on System Synthesis (ISSS)*, pages 84–89.
- [Vahid, 2002] Vahid, F. (2002). *Embedded System Design*. John Wiley & Sons.
- [van de Wiel, 2002] van de Wiel, R. (2002). The code compaction bibliography. *This source has become unavailable, check <http://www.iro.umontreal.ca/~latendre/codeCompression/codeCompression/node1.html> instead.*
- [Vladimirescu, 1987] Vladimirescu, A. (1987). SPICE user's guide. *Northwest Laboratory for Integrated Systems, Seattle*.
- [Vogels and Gielen, 2003] Vogels, M. and Gielen, G. (2003). Figure of merit based selection of A/D converters. *Design, Automation and Test in Europe (DATE)*, pages 1190–1191.
- [Wagner and Leupers, 2002] Wagner, J. and Leupers, R. (2002). Advanced code generation for network processors with bit packet addressing. *Workshop on Network Processors (NPI)*.

- [Wedde and Lind, 1998] Wedde, H. and Lind, J. (1998). Integration of task scheduling and file services in the safety-critical system MELODY. *EUROMICRO '98 Workshop on Real-Time Systems, IEEE Computer Society Press*, page 18pp.
- [Wegener, 2000] Wegener, I. (2000). *Branching programs and binary decision diagrams – Theory and Applications*. SIAM Monographs on Discrete Mathematics and Applications.
- [Weiser, 2003] Weiser, M. (2003). Ubiquitous computing. <http://www.ubiq.com/hypertext/weiser/UbiHome.html>.
- [Weste et al., 2000] Weste, N. H. H., Eshraghian, K., Michael, S., Michael, J. S., and Smith, J. S. (2000). *Principles of CMOS VLSI Design: A Systems Perspective*. Addison-Wesley.
- [Willems et al., 1997] Willems, M., Bürgens, V., Keding, H., Grötter, T., and Meyr, H. (1997). System level fixed-point design based on an interpolative approach. *Design Automation Conference (DAC)*, pages 293–298.
- [Wilton and Jouppi, 1996] Wilton, S. and Jouppi, N. (1996). CACTI: An enhanced access and cycle time model. *Int. Journal on Solid State Circuits*, 31(5):677–688.
- [Wind River Systems, 2003] Wind River Systems (2003). Web pages. <http://www.windriver.com>.
- [Winkler, 2002] Winkler, J. (2002). The CHILL homepage. <http://www1.informatik.uni-jena.de/languages/chill/chill.htm>.
- [Wolf, 2001] Wolf, W. (2001). *Computers as Components*. Morgan Kaufmann Publishers.
- [Wolsey, 1998] Wolsey, L. (1998). *Integer Programming*. Jon Wiley & Sons.
- [Wong et al., 2001] Wong, C., Marchal, P., Yang, P., Prayati, A., Cathoor, F., Lauwereins, R., Verkest, D., and Man, H. D. (2001). Task concurrency management methodology to schedule the MPEG4 IM1 player on a highly parallel processor platform. *9th Intern. Symp. on Hardware/Software Codesign (CODES)*, pages 170–177.
- [Xue, 2000] Xue, J. (2000). *Loop tiling for parallelism*. Kluwer Academic Publishers.
- [Young, 1982] Young, S. (1982). *Real Time Languages –design and development–*. Ellis Horwood.

About the Author

Peter Marwedel



Peter Marwedel was born in Hamburg, Germany. He received his PhD in Physics from the University of Kiel, Germany, in 1974. From 1974 to 1989, he was a faculty member of the Institute for Computer Science and Applied Mathematics at the same University. He has been a professor at the University of Dortmund, Germany, since 1989. He is heading the embedded systems group at the computer science department and is also chairing ICD e.V., a local company specializing in technology transfer. Through ICD, the knowledge about compilers for embedded processors is incorporated into commercial products and is made available to external customers. Dr. Marwedel was a visiting professor of the University of Paderborn in 1985/1986 and of the University of California at Irvine in 1995. He served as Dean of the Computer

Science Department from 1992 to 1995. Dr. Marwedel has been active in making the DATE conference successful and in initiating the SCOPES series of workshops. He started to work on high-level synthesis in 1975 (in the context of the MIMOLA project) and focused on the synthesis of very long instruction word (VLIW) machines. Later, he added compilation for embedded processors (with emphasis on retargetability) to his scope. His projects also include synthesis of self-test programs for processors. His work comprises codesign, energy-aware compilation. Recent work on multimedia-based training led to the design of the RaVi multimedia units (see [//ls12-www.cs.uni-dortmund.de/ravi](http://ls12-www.cs.uni-dortmund.de/ravi)).

Dr. Marwedel is a member of ACM, IEEE CS, and Gesellschaft für Informatik (GI).

He is married and has two daughters and a son. His hobbies include skiing, model railways and photography.

E-mail: peter.marwedel@udo.edu

Web-site: <http://ls12-www.cs.uni-dortmund.de/~marwedel>

List of Figures

0.1	Positioning of the topics of this book	xv
1.1	Influence of embedded systems on ubiquitous computing	5
1.2	SMARTpen	6
1.3	Controlling a valve	7
1.4	Robot “Johnnie” (courtesy H. Ulbrich, F. Pfeiffer, Lehrstuhl für Angewandte Mechanik, TU München), ©TU München	8
1.5	Simplified design information flow	10
2.1	State diagram with exception k	15
2.2	State diagram	18
2.3	Hierarchical state diagram	19
2.4	State diagram using the default state mechanism	20
2.5	State diagram using the history and the default state mechanism	21
2.6	Combining the symbols for the history and the default state mechanism	21
2.7	Answering machine	22
2.8	Answering machine with modified on/off switch processing	22
2.9	Timer in StateCharts	23
2.10	Servicing the incoming line in Lproc	23
2.11	Mutually dependent assignments	25

2.12	Cross-coupled registers	25
2.13	Steps during the execution of a StateCharts model	26
2.14	Symbols used in the graphical form of SDL	31
2.15	FSM described in SDL	31
2.16	SDL-representation of fig. 2.15	31
2.17	Declarations, assignments and decisions in SDL	32
2.18	SDL interprocess communication	32
2.19	Process interaction diagram	33
2.20	Describing signal recipients	33
2.21	SDL block	34
2.22	SDL system	34
2.23	SDL hierarchy	34
2.24	Using timer T	35
2.25	Small computer network described in SDL	35
2.26	Protocol stacks represented in SDL	36
2.27	Single track railroad segment	37
2.28	Using resource “track”	37
2.29	Freeing resource “track”	38
2.30	Conflict for resource “track”	38
2.31	Model of Thalys trains running between Amsterdam, Cologne, Brussels, and Paris	39
2.32	Nets which are not pure (left) and not simple (right)	40
2.33	Generation of a new marking	41
2.34	The dining philosophers problem	43
2.35	Place/transition net model of the dining philosophers problem	43
2.36	Predicate/transition net model of the dining philosophers problem	44
2.37	Message sequence diagram	45

2.38	Railway traffic displayed by a message sequence diagram (courtesy H. Brändli, IVT, ETH Zürich), ©ETH Zürich	46
2.39	Segment from an UML sequence diagram	47
2.40	Activity diagram [Kobryn, 2001]	48
2.41	Use case example	49
2.42	Dependence graph	50
2.43	Task graphs including timing information	51
2.44	Task graphs including I/O-nodes and edges	51
2.45	Task graph including jobs of a periodic task	52
2.46	Hierarchical task graph	53
2.47	Graphical representations of synchronous data flow	54
2.48	An entity consists of an entity declaration and architectures	60
2.49	Full-adder and its interface signals	60
2.50	Schematic describing structural body of the full adder	61
2.51	Outputs that can be effectively disconnected from a wire	64
2.52	Right output dominates bus	64
2.53	Partial order for value set {'0', '1', 'Z', 'X'}	65
2.54	Output using depletion transistor	65
2.55	Partial order for value set {'0', '1', 'Z', 'X', 'H', 'L', 'W'}	66
2.56	Pre-charging a bus	67
2.57	Partial order for value set {'0', '1', 'Z', 'X', 'H', 'L', 'W', 'h', 'l', 'w'}	67
2.58	VHDL simulation cycles	71
2.59	RS-Flipflop	72
2.60	δ cycles for RS-flip-flop	73
2.61	Structural hierarchy of SpecC example	76
2.62	Language comparison	82
2.63	Using various languages in combination	83

3.1	Simplified design information flow	87
3.2	Hardware in the loop	88
3.3	Acceleration sensor (courtesy S. Bütgenbach, IMT, TU Braunschweig), ©TU Braunschweig, Germany	89
3.4	Sample-and-hold-circuit	91
3.5	Flash A/D converter	92
3.6	Circuit using successive approximation	93
3.7	Single-ended signaling	95
3.8	Differential signaling	96
3.9	Hardware efficiency	99
3.10	Dynamic power management states of the StrongArm Processor SA 1100	101
3.11	Decompression of compressed instructions	103
3.12	Re-encoding THUMB into ARM instructions	104
3.13	Dictionary approach for instruction compression	105
3.14	Internal architecture of the ADSP 2100 processor	106
3.15	AGU using special address registers	108
3.16	Wrap-around vs. saturating arithmetic for unsigned integers	109
3.17	Parameters of a fixed-point number system	109
3.18	Using 64 bit registers for packed words	110
3.19	VLIW architecture (example)	111
3.20	Instruction packets for TMS 320C6xx	112
3.21	Partitioned register files for TMS 320C6xx	113
3.22	M3-DSP (simplified)	113
3.23	Branch instruction and delay slots	114
3.24	Floor-plan of Virtex II FPGAs	116
3.25	Virtex II CLB	117
3.26	Virtex II Slice (simplified)	117
3.27	Cycle time and power as a function of the memory size	118

<i>List of Figures</i>	233
3.28 Increasing gap between processor and memory speeds	119
3.29 Memory map with scratch-pad included	119
3.30 Energy consumption per scratch pad and cache access	120
3.31 D/A-converter	121
3.32 Microsystem technology based actuator motor (partial view; courtesy E. Obermeier, MAT, TU Berlin), ©TU Berlin	123
4.1 Simplified design information flow	126
4.2 Classes of scheduling algorithms	128
4.3 Task descriptor list in a TT operating system	129
4.4 Definition of the laxity of a task	131
4.5 EDF schedule	132
4.6 Least laxity schedule	133
4.7 Scheduler needs to leave processor idle	134
4.8 Precedence graph and schedule	135
4.9 Notation used for time intervals	136
4.10 Right hand side of equation 4.5	137
4.11 Example of a schedule generated with RM scheduling	138
4.12 RM schedule does not meet deadline at time 8	138
4.13 EDF generated schedule for the example of 4.12	139
4.14 Priority inversion for two tasks	140
4.15 Priority inversion with potentially large delay	141
4.16 Priority inheritance for the example of fig. 4.15	142
4.17 Real-time kernel (left) vs. general purpose OS (right)	146
4.18 Hybrid OSs	147
4.19 Access to remote objects using CORBA	149
5.1 Simplified design information flow	151
5.2 Platform-based design	152
5.3 Merging of tasks	154
5.4 Splitting of tasks	154

5.5	System specification	155
5.6	Generated software tasks	156
5.7	Memory layout for two-dimensional array $p[j][k]$ in C	159
5.8	Access pattern for unblocked matrix multiplication	161
5.9	Access pattern for tiled/blocked matrix multiplication	162
5.10	Splitting image processing into regular and special cases	163
5.11	Results for loop splitting	165
5.12	Reference patterns for arrays	165
5.13	Unfolded (left) and inter-array folded (right) arrays	166
5.14	Intra-array folded arrays	166
5.15	General view of hardware/software partitioning	168
5.16	Merging of task nodes mapped to the same hardware component	170
5.17	Task graph	175
5.18	Design space for audio lab	177
5.19	Energy reduction by compiler-based mapping to scratch-pad for bubble sort	181
5.20	Comparison of memory layouts	182
5.21	Memory allocation for access sequence (b, d, a, c, d, c) for a single address register A	182
5.22	Reduction of the cycle count by vectorization for the M3-DSP	184
5.23	Possible voltage schedule	187
5.24	Two more possible voltage schedules	187
5.25	Codesign methodology possible with SpecC	191
5.26	Global view of IMEC design flow	192
5.27	Pareto curves for processor combinations 2 and 3	193
5.28	COSYMA design flow	194
6.1	Simplified design information flow	200
6.2	Scan path design	203

<i>List of Figures</i>	235
6.3 BILBO	204
6.4 Segment from processor hardware	205
6.5 Fault tree	208
6.6 Inputs for model checking	210

Index

- A/D-converter, 91
- ACID-property, 148
- actor, 17
- actuator, 2, 122
- ADA, 55
- address generation unit, 108, 183
- address register, 182
- ambient intelligence, 5, 99, 232
- API, 125, 131, 147
- application domains, 15
- application-specific circuit (ASIC), 98, 100
- arithmetic
 - fixed-point ~, 109, 157
 - floating-point ~, 157
 - saturating ~, 108, 109
- ARM, 104
- artificial eye, 90
- ASIC, 98, 100, 115
- availability, 2
- basic block, 195
- behavior
 - deterministic ~, 18, 25, 28, 72, 74
 - non-deterministic ~, 27
 - non-functional ~, 16
 - real-time ~, 96
- BILBO, 204
- Binary Decision Diagram (BDD), 81, 209, 210
- Bluetooth, 98
- boundary scan, 204
- branch delay penalty, 114, 185
- broadcast, 26, 28, 30, 79
- building
 - smart ~, 1, 7
- cache, 110, 119
- CACTI cache estimation tool, 120
- CardJava, 59
- causal dependence, 50
- channel, 17, 33
- charge-coupled devices (CCD), 89
- Chill, 78
- clock synchronization, 145
- code size, 2, 181, 183
- communication, 15, 29, 93
 - blocking ~, 55
 - non-blocking ~, 29
- compiler, 177
 - energy-aware ~, 178
 - for digital signal processor, 181
 - retargetable ~, 178, 185
- composability, 144
- compression
 - dictionary-based ~, 105
- computer
 - disappearing ~, xi, 1, 3
- computing
 - pervasive ~, 1, 5
 - ubiquitous ~, 5
- concurrency, 15
- condition/event net, 40
- configurability, 143
- configuration
 - link-time ~, 146
- context switch, 143, 154
- contiguous files, 145
- controller area network (CAN), 97
- COOL, 167, 168, 195
- cost, 3, 173
 - estimated ~, 127
 - function for scheduling, 130
 - function of integer programming, 176
 - model for energy, 179
 - model of COOL, 169
 - model of integer programming, 171
 - of ASICs, 100
 - of CCDs, 89
 - of communication, 94

- of damages, 207
 - of energy, 100
 - of floating point arithmetic, 109
 - of second instruction set, 105
 - of testing, 205
 - of wiring, 97
- COSYMA, 194
- coverage, 202
- critical section, 29, 140
- CSA-theory, 63
- CSMA/CA, 97
- CSP, 55, 196
- CTL, 210
- curriculum, xii
- cyclic redundancy check (CRC), 204

- D/A-converter, 121
- damage, 207
- dataflow, 17
 - synchronous ~, 18, 196
- deadline, 30, 128, 130, 132, 136, 138, 139, 145
- deadline interval, 131, 135
- DECT, 98
- dependability, 2, 14, 83, 145, 207
- dependence graph, 50
- depletion transistor, 65
- design flow, 10, 87, 151, 200
- design for testability, 202
- design space estimation, 190
- design space exploration, 192
- diagnosability, 95
- diagrams
 - of UML, 47
- differential signaling, 95
- dining philosophers problem, 44
- discrete event, 17, 196
- dispatcher, 129
- dynamic power management (DPM), 189
- dynamic voltage scaling (DVS), 101, 102, 186, 187, 192

- eCos, 143
- EDF, 139
- efficiency, 2, 14, 94
 - code-size ~, 102
 - energy ~, 2, 101, 119
 - run-time ~, 3, 105
- electro-magnetic compatibility (EMC), 200
- embedded system(s), 1
 - hardware, 87
 - market of ~, 8
- Embedded Windows XP, 148
- energy, 2, 99, 178
- EPIC, 111
- Estelle, 78
- Esterel, 28, 79, 196
- European installation bus (EIB), 98

- event, 14, 36, 40, 71, 79
- exception, 14, 20, 21, 58, 78
- executability, 15

- failure mode and effect analysis (FMEA), 208
- fault
 - injection, 207
 - model, 202, 205
 - simulation, 206
 - tolerance, 94
 - tree, 208
 - tree analysis (FTA), 207
- field programmable gate arrays (FPGAs), 116, 194, 201
- FIFO, 17, 53, 54, 78
 - in SDL, 32
- finite state machine (FSM), 16, 18, 20, 30, 31, 196, 210
 - communicating ~, 17
- formal verification, 209

- garbage collection, 58, 145
- gated clocking, 101
- granularity, 52, 195

- hardware description language, 59
- hardware in the loop, 88
- hardware/software codesign, 151
- hardware/software partitioning, 167, 190, 195
- hazard, 207
- hierarchy, 13
 - in SDL, 33
 - in StateCharts, 19
 - leaf, 169
 - leaves, 20, 34
- history mechanism, 20
- homing sequence, 203

- IEC60848, 78
- IEEE 1076, 59
- IEEE 1164, 62
- IEEE 1364, 75
- IEEE 802.11, 98
- IMEC, 191
- inlining, 185
- input, 14, 16, 19, 27, 30, 32, 51, 53–55, 60
- input/output, 30
- instruction level parallelism, 183
- instruction set architecture (ISA), 80
- instruction set level, 80
- integer programming, 170, 171, 181, 188, 189
- intellectual property, 125
- interrupt, 144, 145
- ITRON, 146

- Java, 58, 145
- job, 128, 135
- JTAG, 204

- Kahn process network, 53, 196
- knapsack problem, 181
- lab, xiv
- language, 13
 - synchronous ~, 27
- laxity, 131, 135
- LDF, 134
- locality, 161
- logic
 - first-order ~, 209
 - higher order ~, 210
 - multi-valued ~, 62
 - propositional ~, 209
 - reconfigurable ~, 115
- loop
 - blocking, 160
 - fission, 160
 - fusion, 160
 - permutation, 159
 - splitting, 163
 - tiling, 160
 - unrolling, 160
- LOTOS, 78
- maintainability, 2, 95
- MAP, 98
- marking, 41
- MATLAB, 79, 80
- maximum lateness, 130
- memory, 118
 - bank, 110
 - hierarchy, 180
 - layout, 182
- message passing, 28
 - asynchronous ~, 17, 29
 - synchronous ~, 18
- message sequence charts (MSC), 44
- microcontroller, 115
- middleware, 125
- MIMOLA, 59
- model
 - discrete event ~, 17
 - layout level ~, 82
 - of computation, 16
 - switch-level ~, 81
- module chart, 27
- MSC, 44, 45
- multi-thread graph, 52
- multiply/accumulate instruction, 110
- mutex primitives, 140
- mutual exclusion, 37, 51, 145
- NP-hard, 209
- object orientation, 15, 196
- occam, 55
- OCTOPUS, 196
- open collector circuit, 63
- operating system
 - driver, 143
 - kernel, 146
 - real-time ~, 125, 144
- optimization, 163, 168, 170–172, 178–181, 183–186, 190, 193
 - high-level ~, 157
- OSEK, 146
- Pareto curves, 192
- Pearl, 78
- period, 128
- periodic schedules, 52
- Petri net, 36, 155
- place/transition net, 40
- platform-based design, 87, 125
- portability, 16
- post-PC era, xi, 9
- power, 99, 178
- power models, 179
- pre-charging, 66
- pre-requisites, xii
- predecessor, 50
- predicate/transition net, 42
- predicated execution, 113, 185
- predictability, 126, 130, 140, 144, 149
- prefetching, 161
- priority ceiling protocol, 143
- priority inheritance, 141
- priority inversion, 140
- privacy, 95
- processes, 28
- processor, 100, 178
 - DSP~, 108
 - multimedia ~, 110, 184
 - network ~, 185
 - very long instruction word (VLIW) ~, 111
 - VLIW ~, 184
- program
 - self-test ~, 205
- protection, 143
- Ptolemy, 195
- rapid prototyping, 201
- readability, 15
- real-time, 58
 - behavior, 94
 - capability, 110
 - constraint, 3
 - CORBA, 149
 - data bases, 125, 148
 - hard ~ constraint, 4
 - kernel, 145
 - POSIX, 150

- real-time operating system (RTOS), 10, 127, 143–146, 148, 156
- register file, 110, 112, 183
- register-transfer level, 81
- reliability, 2
- rendez-vous, 29, 56
- resolution function, 64
- resource allocation, 147
- robotics, 7, 10
- robustness, 94, 95
- Rosetta, 78
- row major order, 159, 161
- RTOS, 144

- safety, 2, 83, 143
- safety case, 208
- sample-and-hold circuit, 90
- scan design, 202
- scan path, 203
- schedulability tests, 130
- scheduling, 127, 128, 145
 - dynamic ~, 129
 - earliest deadline first ~, 139
 - instruction ~, 180
 - least laxity ~, 133
 - non-preemptive ~, 128
 - optimal ~, 135
 - rate monotonic ~, 136
- scratch pad memory (SPM), 119, 181
- SDF, 54, 196
- SDL, 30
- security, 2, 143
- select-statement, 57, 68
- semantics
 - SDL ~, 32
 - StateChart ~, 24
 - VHDL ~, 69
- sensor, 2, 88
 - bio-metrical ~, 90
 - image ~, 89
- sequence diagram, 47
- shared memory, 28
- signal-to-noise-ration (SNR), 158
- signaling
 - differential ~, 96
 - single-ended ~, 95
- Silage, 78
- SIMD-instructions, 110
- simulation, 200
 - bit-true ~, 80
 - cycle-true ~, 81
- Simulink, 79
- slack, 131, 135
- slides, xiv
- SoC, 3, 103
- SpecC, 76, 190
- SpecCharts, 78
- specification languages, 13
- sporadic task server, 140
- state
 - ancestor ~, 19
 - AND-super ~, 21
 - basic ~, 19
 - default ~, 20
 - diagram, 15, 18
 - OR-super ~, 19
 - super ~, 19
- StateCharts, 18
- STEP 7, 78
- stuck-at-fault, 202
- successive approximation, 92
- successor, 50
- synchronization, 15, 28
- system
 - dedicated ~, 3
 - embedded ~, 5
 - hybrid ~, 4
 - reactive ~, 4, 196
 - time triggered ~, 129
- system level, 80
- system on a chip (SoC), 3, 58, 103, 167, 190
- SystemC, 73, 80
- SystemVerilog, 75

- task
 - aperiodic ~, 128
 - concurrency management, 152, 153, 192
 - periodic ~, 128, 135
 - sporadic ~, 128, 130
- task graph, 50
 - node splitting, 154
- termination, 16
- test, 201
- testability, 202
- THUMB, 104
- time, 45, 50, 60, 69, 74, 78
- time services, 145
- timer, 23, 34
 - in SDL, 35
- timing, 29
- timing behavior, 14
- timing information, 50
- transaction level modeling, 80

- UML, 45
- unified modeling language, 45
- user-interface, 3

- validation, 199
- variable voltage processor, 188
- Verilog, 75
- VHDL, 25, 59
 - architecture, 60
 - entity, 60
 - port map, 62

signal driver, 64
VHDL-AMS, 80
VxWORKS, 146

WCET, 126
weight, 3

Wind River Systems, 148
Windows CE, 148
worst-case execution time, 126

Z language, 78
zero-overhead loop instruction, 107, 160, 184