# An Annotated Embedded Systems Bibliography

Stephen A. Edwards
Computer Science Department
Columbia University

October 23, 2001

## 1 Books

Barr [4] is a readable general introduction that focuses on mundane issues such as compilers, linkers, and target boards.

Brown [12] is another very practical book, discussing such issues as compilers, linking, and interrupt latency. The book contains an extensive example of a pilot control panel.

Simon [38] is more comprehensive than Barr, speaking of RTOSes.

Burns and Wellings [14] is an academic text aimed at graduate students.

Grehan and Cyliax [25] talks more about RTOSes.

Ganssle [24] is a broad survey, covering issues such as which tools to buy. Probably fairly dated by now.

Wolf's textbook [41] is aimed at undergraduates.

Bentham [5] talks about implementing TCP/IP, both for "normal" systems and finally an incredibly stripped-down version that runs on a tiny 8-bit PIC microcontroller.

Thomas and Moorby [40] is the standard reference on Verilog, a widely-used language for modeling and specifying digital hardware.

Ellsberger et al. [19] is the most readable book on SDL, a graphical language used mostly in Europe to describe telecommunication protocols.

The first two chapters of Thoen and Catthoor [39] give a reasonable overview of some embedded systems issues. The rest of the book is devoted to their Petri-net-like model for embedded software.

Gajski et al. [21] resembles Thoen and Catthoor [39] in that the early chapters are a reasonable survey of what's going on in the field. The rest describe their particular little-used modeling language. A later book [22] describes another of their hardware/software modeling languages.

Balarin et al. [3] is a complete reference on POLIS, an academic tool designed mostly for the synthesis of automotive controllers that laid the foundation for Cadence's commercial VCC tool.

Gallmeister [23] describes the POSIX.4 standard, the real-time extensions to the Unix-centric POSIX standards. It's rather a heavyweight interface, not

clearly useful for all applications, but gives some insight into some things found in RTOSes.

Rubini [36] describes how to write Linux device drivers. Most embedded systems both use existing and require new device drivers of some sort.

Shaw [37] focuses on real-time systems, discussing issues such as scheduling, RTOSes, clock distribution, and execution time prediction. His motivating example is an air-traffic control system.

Labrosse [31] describes his MicroC/OS-II RTOS in excruciating detail. If you ever wondered how to implement a preemptive priority-based scheduler, this is the book to read.

# 2 Periodicals

Circuit Cellar magazine (www.circuitcellar.com) is a monthly magazine containing hobbyist projects that are usually embedded systems-like. One problem: they speak mostly about the design of the hardware and assume that you'll just go and download the software.

Embedded Systems Programming (www.embedded.com) is a trade magazine for embedded systems developers. It usually has some lightweight discussions of programming, real-time issues, and whatnot. Its view of embedded systems is off-the-shelf single-board computers running an RTOS.

Kluwer publishes Design Automation for Embedded Systems, a scholarly journal on embedded systems issues.

# 3 Companies

Wind River Systems (www.windriver.com) is the main supplier of RTOSes (VxWorks) and and IDE and development tools surrounding it.

TI (www.ti.com) is the leading supplier of DSPs.

Motorola (www.motorola.com) is another big supplier of embedded microprocessors, such as the 68000 series.

# 4 Papers

General surveys [15, 18]. An RTOS survey [34].

Polis-related: Verifying a CFSM network [2]. Synthesizing software from CFSMs [17, 1]. The shock-absorber example [16].

Synthesizing dataflow graphs [6, 11, 7, 9, 10, 8].

The El Greco project, which turned into Synopsys' CoCentric System Studio tool for modeling dataflow and control [13].

The multi-language approach [20].

Edward's view of what should be happening [32, 33].

Performance estimation [35].

Hermann Kopetz has long been promoting his Time-Triggered Protocol for communications within such real-time systems as cars. He's publishes extensively on the idea. A broad survey: [27]. The philosophy: [28, 29]. How to synchronize the clocks in TTP: [30]. Modeling time: [26].

# References


[1] Felice Balarin, Massimiliano Chiodo, Paolo Giusto, Harry Hsieh, Attila Jurecska, Luciano Lavagno, Alberto Sangiovanni-Vincentelli, Ellen M. Sentovich, and Kei Suzuki. Synthesis of software programs for embedded control applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):834–849, June 1999.

[2] Felice Balarin, Massimiliano Chiodo, Luciano Lavagno, Harry Hsieh, Alberto Sangiovanni-Vincentelli, Paolo Giusto, and Attila Jurecska. Formal verification of embedded systems based on CFSM networks. In *CODES 96*, 1995.

[3] Felice Balarin, Paolo Giusto, Attila Jurecska, Claudio Passerone, Ellen Sentovich, Bassam Tabbara, Massimiliano Chiodo, Harry Hsieh, Luciano Lavagno, Alberto Sangiovanni-Vincentelli, and Kei Suzuki. *Hardware-Software Co-Design of Embedded Systems: The POLIS Approach*. Kluwer, Boston, Massachusetts, 1997.

[4] Michael Barr. *Programming Embedded Systems in C and C++*. O'Reilly & Associates, Inc., Sebastopol, California, 1999.

[5] Jeremy Bentham. *TCP/IP Lean: Web Servers for Embedded Systems*. CMP Books, Lawrence, Kansas, 2000.

[6] Shuvra S. Bhattacharyya, Joe T. Buck, Soonhoi Ha, Praveen K. Murthy, and Edward A. Lee. A scheduling framework for minimizing memory requirements of multirate DSP systems represented as dataflow graphs. In *Proceedings of the IEEE Workshop on VLSI Signal Processing VI*, pages 188–196, Veldhoven, The Netherlands, October 1993.

[7] Shuvra S. Bhattacharyya and Edward A. Lee. Looped schedules for dataflow descriptions of multirate signal processing algorithms. *Journal of Formal Methods in System Design*, 5(3):183–205, December 1994.

[8] Shuvra S. Bhattacharyya, Ranier Leupers, and Peter Marwedel. Software synthesis and code generation for signal processing systems. *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, 47(9):849–875, September 2000.

[9] Shuvra S. Bhattacharyya, Praveen K. Murthy, and Edward A. Lee. *Software Synthesis from Dataflow Graphs*. Kluwer, Boston, Massachusetts, 1996.

[10] Shuvra S. Bhattacharyya, Praveen K. Murthy, and Edward A. Lee. Synthesis of embedded software from synchronous dataflow specifications. *Journal of VLSI Signal Processing Systems*, 21(2):151–166, June 1999.

[11] Shuvra Shikhar Bhattacharyya. *Compiling Dataflow Programs for Digital Signal Processing*. PhD thesis, University of California, Berkeley, July 1994. Available as UCB/ERL M94/52.

[12] John Forrest Brown. *Embedded Systems Programming in C and Assembly*. Van Nostrand Reinhold, New York, New York, 1994.

[13] Joseph Buck and Radha Vaidyanathan. Heterogeneous modeling and simulation of embedded systems in El Greco. In *Proceedings of the Eighth International Workshop on Hardware/Software Codesign (CODES)*, San Diego, California, May 2000.

[14] Alan Burns and Andy Wellings. *Real-Time Systems and Programming Languages*. Addison-Wesley, Reading, Massachusetts, second edition, 1997.

[15] Raul Camposano and Jörg Wilberg. Embedded system design. *Design Automation for Embedded Systems*, 1:5–50, 1996.

[16] Massimiliano Chiodo, Daniel Engels, Paolo Giusto, Harry Hsieh, Attila Jurecska, Luciano Lavagno, Kei Suzuki, and Alberto Sangiovanni-Vincentelli. A case study in computer-aided co-design of embedded controllers. *Design Automation for Embedded Systems*, 1(1):51–67, January 1996.

[17] Massimiliano Chiodo, Paolo Giusto, Attila Jurecska, Luciano Lavagno, Harry Hsieh, Kei Suzuki, Alberto Sangiovanni-Vincentelli, and Ellen Sentovich. Synthesis of software programs for embedded control applications. In *Proceedings of the 32nd Design Automation Conference*, pages 587–592, San Francisco, California, June 1995.

[18] Stephen Edwards, Luciano Lavagno, Edward A. Lee, and Alberto Sangiovanni-Vincentelli. Design of embedded systems: Formal models, validation, and synthesis. *Proceedings of the IEEE*, 85(3):366–390, March 1997.

[19] Jan Ellsberger, Dieter Hogrefe, and Amardeo Sarma. *SDL: Formal Object-Oriented Language for Communicating Systems*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 1997.

[20] Rolf Ernst and Ahmed Amine Jerraya. Embedded system design with multiple languages. In *Proceedings of the Asia South Pacific Design Automation Conference (ASP-DAC)*, pages 391–396, Yokohama, Japan, January 2000.

[21] Daniel D. Gajski, Frank Vahid, Sanjiv Narayan, and Jie Gong. *Specification and Design of Embedded Systems*. Prentice Hall, 1994.

[22] Daniel D. Gajski, Jianwen Zhu, Rainer Dömer, Andreas Gerstlauer, and Shuqing Zhao. *SpecC: Specification Language and Methodology*. Kluwer, Boston, Massachusetts, 2000.

[23] Bill O. Gallmeister. *POSIX.4: Programming for the Real World*. O'Reilly & Associates, Inc., Sebastopol, California, 1995.

[24] Jack G. Ganssle. *The Art of Programming Embedded Systems*. Academic Press, San Diego, 1992.

[25] Moore Grehan and Cyliax. *Real-Time Programming; A Guide to 32-Bit Embedded Development*. aw, Reading, Massachusetts, 1998.

[26] Hermann Kopetz. Sparse time versus dense time in distributed real-time systems. In *Proceedings of the 12th International Conference on Distributed Computing Systems*, pages 460–467, Yokohama, Japan, June 1992. IEEE Press.

[27] Hermann Kopetz. *The Future of Software Engineering*, chapter Software Engineering for Real-Time: A Roadmap, pages 201–211. ACM Press, 2000.

[28] Hermann Kopetz, Martin Braun, Christian Ebner, Andreas Kruger, Dietmar Millinger, Roman Nossal, and Anton Schedl. The design of large real-time systems: The time-triggered approach. In *Proceedings of the 16th IEEE Real-Time Systems Symposium (RTSS)*, pages 182–189, Pisa, Italy, December 1995.

[29] Hermann Kopetz and Günter Grünsteidl. TTP — a protocol for fault-tolerant real-time systems. *IEEE Computer*, 27(1):14–23, January 1994.

[30] Hermann Kopetz and Wilhelm Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, C–36(8):933–940, August 1987.

[31] Jean Labrosse. *MicroC/OS-II*. CMP Books, Lawrence, Kansas, 1998.

[32] Edward A. Lee. Embedded software—an agenda for research. Technical Report UCB/ERL M99/63, University of California, Berkeley, December 1999.

[33] Edward A. Lee. What's ahead for embedded software? *IEEE Computer*, 33(9):18–26, September 2000.

[34] Yanbing Li, Miodrag Potkonjak, and Wayne Wolf. Real-time operating systems for embedded computing. In *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, pages 388–392, Austin, Texas, October 1997.

[35] Sharad Malik, Margaret Martonosi, and Yau-Tsun Steven Li. Static timing analysis of embedded software. In *Proceedings of the 34th Design Automation Conference*, June 1997.

[36] Alessandro Rubini. *Linux Device Drivers*. O'Reilly & Associates, Inc., Sebastopol, California, 1998.

[37] Alan C. Shaw. *Real-Time Systems and Software*. John Wiley & Sons, New York, 2001.

[38] David E. Simon. *An Embedded Software Primer*. Addison-Wesley, Reading, Massachusetts, 1999.

[39] Filip Thoen and Francky Catthoor. *Modeling, Verification, and Exploration of Task-Level Concurrency in Real-Time Embedded Systems*. Kluwer, Boston, Massachusetts, 2000.

[40] Donald E. Thomas and Philip R. Moorby. *The Verilog Hardware Description Language*. Kluwer, Boston, Massachusetts, fourth edition, 1998.

[41] Wayne Wolf. *Computers as Components: Principles of Embedded Computer Systems Design*. Morgan Kaufmann, San Francisco, California, 2000.

Examples of Annotated Bibliography APA. An annotated bibliography is a list where you gather all the references that you have used on your paper. An annotated bibliography has to include a comment or critical evaluation about each source. You can read some examples of annotated bibliography APA to see how it has to be done. An annotated bibliography APA is important because it helps to demonstrate your critical thinking skills. It is a way to prove that you have researched the subject and that you know what you are talking about. An APA annotated bibliography is a reference list that follows t