Guest Editor’s Introduction:

The Status of High-Level Synthesis

**High-level synthesis** (occasionally called behavioral synthesis) is the design task of converting an abstract behavioral description of a digital system into a register-transfer level design implementing that behavior. The behavioral description specifies the functionality of the digital system to be designed at a high level, usually in either a procedural programming language, such as C or Pascal, or in a hardware description language, such as VHDL or Verilog. From this behavioral description, the high-level synthesis system generates a structural design implementing that behavior. Composing the design are such register-transfer level components as arithmetic logic units (ALUs), dedicated functional units (adders or multipliers, for example), multiplexers, buses, and registers.

High-level synthesis has been an area of active research for 20 years or more, both in industry and academia. *IEEE Design and Test, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* and *IEEE Transactions on VLSI Systems* cover high-level synthesis. Numerous conferences, including the Design Automation Conference (DAC), the International Conference on Computer-Aided Design (ICCAD), the European Design and Test Conference (ELTUC, formerly EDAC), the European Design Automation Conference (EuroDAC), and the International Symposium on High-Level Synthesis (ISHLS), feature papers on the subject. Several publishers offer textbooks and surveys on it as well.

The field of high-level synthesis is more exciting now than ever. Engineers have designed real chips for commercial products with in-house, high-level synthesis tools. Companies are commercially marketing high-level synthesis tools. Researchers are pushing forward in bold new directions—system-level synthesis, hardware-software codesign, and synthesis for low power, testability, and fault tolerance, among others. After years of trying to understand the basic issues, we now have a handle on those basics, and are beginning to develop ever more useful tools for industry. This four-part special series will examine the current status of high-level synthesis and point toward future directions for the field.

The first article in the series, “Introduction to High-Level Synthesis,” by Daniel Gajski and Loganath Ramachandran appears in this issue. Their article places high-level synthesis into perspective in the design process and defines its basic problems.

The second article, “The Scheduling Problem,” by Robert Walker and Samit Chaudhuri, will appear in the Summer 1995 issue. It examines the scheduling problem—one of the two central tasks in high-level synthesis. Scheduling is the task of determining the sequence in which to execute the operators. The result of scheduling is a control step schedule, specifying which operations execute in each control step, or state. This tutorial will define the various scheduling problems and explain the scheduling algorithms commonly used in high-level synthesis today.

The third article, “The Data-Path Synthesis Problem,” will appear in the Fall 1995 issue. That problem is the second of the two central tasks in high-level synthesis. Data-path synthesis consists of allocation (setting aside the appropriate number of functional, storage, and interconnection units) and bind-
ing (assigning operations to functional units and values to storage units, as well as interconnecting those components). Allocation and binding result in a register-transfer level data path. Like the scheduling tutorial, this tutorial will define the various data-path synthesis problems and explain the more common data path synthesis algorithms.

**Finally, the series will conclude** in the Winter 1995 issue with "Future Directions," a set of predictions for the future of high-level synthesis made by recognized experts from industry and academia.

---

**Robert A. Walker** is an assistant professor of computer science at Rensselaer Polytechnic Institute. He is the coauthor of two books on high-level synthesis and is particularly interested in those problems connected with scheduling. He has served on the Advisory Board of the ACM SIGDA, has been actively involved with their university booth program, and also serves as SIGDA secretary-treasurer. Walker received the MS and PhD degrees from Carnegie Mellon University. He is a senior member of the IEEE, and a member of the Computer Society, the ACM, ACM SIGDA, and Sigma Xi.

Address correspondence about this series to Robert A. Walker at Rensselaer Polytechnic Institute, Computer Science Department, Troy, NY 12180; walkerb@cs.rpi.edu.