Introduction to the Configware Minitrack
Hardware and Software come closer

Reiner W. Hartenstein
Department of Computer Science
University of Kaiserslautern
Erwin-Schrödinger-Straße, D-67663 Kaiserslautern, Germany
Email: hartenst@rhrk.uni-kl.de
URL: http://xputers.informatik.uni-kl.de

Hardware design methods come closer to software engineering. Proven techniques are further ahead than common awareness of them. Behavioral Synthesis means hardware compilation rather than CAD: Hardware compilation from register-transfer level sources or even from high-level programs like C sources. The programmer “programs” by himself the hardware he needs.

This has become reality by novel dynamically reconfigurable hardware platforms (see papers by Becker/Schmidt, Eisenring/Teich). The paper by Postula et al. shows, that such “programming the hardware” is much more efficient than classical hardware design and even High-level Synthesis.

The papers of this minitrack introduce methods to use such platforms for “programming” the hardware, as well as related application development tools (see paper by Gardner et al.). Dramatic speedups have been obtained by software migration onto such platforms (also see paper by Suyama et al.).

From “classical” compilation techniques, including parallelizing compilation, a rich methodology is available for adaptation to the hardware compilation problem and even the hardware/software partitioning problem, what is demonstrated by solutions introduced by Becker/Schmidt, by Eisenring/Teich, and by Postula et al.).

The microprocessor is no longer general purpose. Most of the silicon real-estate in a workstation, a PC, or even an advanced hand-held device, is dedicated: Accelerators for graphics, multimedia, communication, and many other high-performance functionality. This has a dramatic impact on the implementation of data-processing applications. For an increasing number of applications, the central issue is a partitioning problem for an optimum hardware/software trade-off. Solutions are introduced in the papers by Becker/Schmidt, by Eisenring/Teich and by Postula et al.

Fundamentally new architectures of reconfigurable circuits will revolutionize structural programming (see paper by Becker/Schmidt). An increasing number of analysts and panelists predict, that in the next decade Configware will advance from a niche-technology to mainstream: From tinkertoy approach to a structural computing-paradigm - an alternative to sequential programming.