# The Changing Role of Computer Architecture Education within CS Curricula

(invited presentation)

Reiner Hartenstein
TU Kaiserslautern
http://hartenstein.de

More than 98% of all microprocessors are found within embedded systems. The British Department of Trade and Industry predicts, that by the year 2010 more than 90% of all program code will be implemented for embedded applications. But the qualification of our „typical“ CS (and CIT) graduates is torpedoed by deficits of our current computer architecture education limiting the horizon to procedural programming in the time domain. However, meanwhile the fundamental machine model is no more just the „von Neumann“ paradigm merely supporting an instruction-stream-based mind set. For embedded systems the basic model is a symbiosis between CPU and primarily data-stream-based accelerator co-processors. Implementing applications for embedded systems also requires hardware / software partitioning decisions. Since meanwhile morphware [1] [2] and Reconfigurable Computing (RC) has become mainstream, also the accelerators are programmable by loading configware code downto their hidden RAM [3]. What is urgently needed is the qualification for programming in time (programming software) and programming in space (programming configware). But currently the software for the CPU is mainly implemented by software people, whereas the accelerators are implemented by EEs or other hardware people.

Communication problems between these two groups of experts having different backgrounds are the reason for the deep chasm between RC and the way, how "classical" CS people look at parallelism [4]. The situation is comparable to the well-known hardware / software chasm. In education until recently RC has been mainly the subject of embedded systems or SoC design within EE departments, whereas most classical CS departments have ignored the enormous additional speed-up opportunities which can be obtained from this field. Only a few departments provide special courses mostly attended by a small percentage of graduate students. Conferences like ISCA have stubbornly refused to include RC and related areas in their scope. Also many major players in the IT market have mainly ignored this area.

More recently this situation is beginning to change. An increasing number of colleagues from the area of computer architecture as well as from classical parallel computing or supercomputing communities is going to be ready to discuss fundamental issues with us [5] [6] [7]. Last year, Intel Research at Hillsboro, Oregon, held a major internal workshop on RC. It has been told, that also Microsoft has held an internal workshop on this area. Other major players already joined this movement, like Hewlett Packard, IBM, infineon, Motorola, Sony, ST microelectronics, Texas Instruments, Toshiba, and many others. Accordingly a major break-through also in CS education and CAE is overdue. All scientific know-how ingredients needed are available - ready to be integrated in CS curricula: software / configware co-compilation [8] [9], software to configware migration [10] [11], mapping applications onto morphware [10] [11] [12] [13], architectural resources for data-stream-based anti machines [14] [15], and many others. Not only FPGAs, but also coarse grain data path array platforms are available commercially, along with application development tools [16].

It is time to take these promising opportunities to upgrade our CS curricula by converting programming and software engineering into a duality of software engineering and configware engineering, based on the co-existence of two machine paradigms, the classical instruction-stream-centered model of the CPU, and, the data-stream-based anti machine model, being the direct counterpart of von Neumann. Because of this duality of basic models the configware / software chasm can be easily bridged without requiring hardware and circuit expertise from our CS graduates.

This new road map is based on the duality of an instruction-stream-based mind set, and a data-stream-based mind set [1] [4] [15] [17]. Not only the HPC community urgently needs to benefit from a curricular revision, but also the rapidly increasing percentage of programmers implementing code for embedded systems. However, most CS graduates are not qualified for this changing labour market. With their procedural-only mind set they cannot cope with hardware /

configware / software partitioning. To avoid a disaster for future CS graduates looking for their first job, CS departments have to wake up. Here we have a good chance to become successful trailblazers by forming a RC old boys' network together with colleagues from "classical CS", organized like the Mead & Conway movement more than 20 years ago [18].

# References


[1] R. Hartenstein (invited chapter): Morphware; in: A. Zomaya (editor): Handbook of Innovative Computing; LNCS series, Springer Verlag Heidelberg/New York, 2004

[2] http://morphware.net/

[3] http://configware.org/

[4] R. Hartenstein (invited paper): The Digital Divide of Computing; Proc. 2004 ACM Int'l Conf. on Computing Frontiers (CF04); Ischia, Italy, April 2004

[5] R. Hartenstein (keynote): Software or Configware? About the Digital Divide of Computing; IPDPS 2004, Santa Fe, NM, 2004

[6] R. Hartenstein (opening keynote): Reconfigurable HPC: torpedoed by Deficits in Education? 1st Workshop on Reconfigurable High Performance Computing (RHPC); 7th Int'l Conf. on High Performance Computing and Grid in Asia Pacific Region (HPC Asia 2004), Omiya Sonic City, Japan July 20-22, 2004

[7] R. Hartenstein (invited paper): Data-Stream-based Computing and Morphware; Joint 33rd Speedup and 19th PARS Workshop (Speedup / PARS 2003), Basel, Switzerland, March 2003

[8] K. Schmidt et al.: Combining Structural and Procedural Programming by Parallelizing Compilation; Proc. 1995 ACM Symp. on Applied computing, Nashville, Tenn., Feb 1995

[9] J. Becker et al.: Parallelization in Co-Compilation for Configurable Accelerators; Proc. ASP-DAC'98, Yokohama, Japan, Febr 1998

[10] R. Kress et al.: A Data Path Synthesis System for the reconfigurable Data Path Architecture; Proc. ASP-DAC 1995, Chiba, Japan, August 1995

[11] U. Nageldinger et al: KressArray Xplorer: a new CAD Environment to optimize Reconfigurable Data Path Arrays; Proc. ASP-DAC 2000, Yokohama, Japan, Jan 2000

[12] R. Hartenstein (embedded tutorial): A decade of Reconfigurable Computing: a Visionary Retrospective; Proc. DATE 2002, Munich, Germany, March 2002

[13] R. Hartenstein (embedded tutorial): Coarse Grain Reconfigurable Architectures; Proc. ASP-DAC 2001, Yokohama, Japan, Jan 2001

[14] A. Hirschbiel et al.: A Novel Paradigm of Parallel Computation and its Use to Implement Simple High Performance Hardware; Proc. InfoJapan'90, Tokyo, Japan, 1990 -- Invited reprint in: Future Generation Computer Systems 7 91/92, p. 181-198, North Holland

[15] M. Herz et al. (invited paper): Memory Organization for Data-Stream-based Reconfigurable Computing; Proc. IEEE ICECS 2002, Dubrovnik, Croatia, Sept 2002

[16] http://pactcorp.com

[17] J. Becker et al. (solicited paper): Configware and morphware going mainstream; Journal of Systems Architecture, vol. 49, Issue 4-6 (Sept 2003)

[18] R. Hartenstein (opening keynote): Are we ready for the Breakthrough?; 10th Reconfigurable Architectures Workshop 2003 (RAW 2003), Nice, France, April 2003