# Overdue CS, CE and ECE Curriculum Innovations

Reiner Hartenstein, January 2006 - TU Kaiserslautern, http://hartenstein.de

„Rising crescendos of voices sound alarm about the downturn in young people selecting a computing field for their careers“ [1] [2]. Europe and the U.S. are losing lead in science and engineering [3] [4]. Half a century of U.S. dominance may be slipping as America's share of graduates falls after years of declining enrolment in CS and related programs [5]. Young people find molecular biology more fascinating and believe, that CS-related curricula are obsolete. To make CS more attractive it has been proposed to to teach to the students how to do innovation by CS application [1]. But it's more important to teach professors and accreditation people how to innovate the structure of the curriculum. A key issue is the persistance of the image, that CS is just programming [6] and software engineering. But in fact - and still ignored by most curriculum recommendations - we now have the dichotomy of two programming paradigms: procedural programming for instruction-stream-based software engineering, versus structural programming for configware engineering in Reconfigurable Computing (the inevitable ingredient of embedded systems), which is not instruction-stream-based.

**Most programmers write embedded applications.**

| keyword | Google |
|---|---|
| FPGA | 9,960,000 |
| Reconfigurable Computing | 256,000 |
| Configware | 13,400 |
| (Kress/Kung) Anti Machine | 18,800 |

Fig. 1. (Jan. 2006) found almost 10 million times

**Pervasiveness of Reconfigurable Computing: one click in http://fpl.org/pervasiveness.html shows you the astonishingly high number of hits**

**Typical CS graduates are not well qualified** for such contemporary labor market requirements, where 99% of all microprocessors are used within embedded systems. Embedded software code doubles every 10 months [7]. Most programmers write embedded applications. Unable to understand FPGA[1] application and to decide software / configware / hardware partitioning most CS graduates are not really qualified.

**Years ago, reconfigurable platforms went mainstream in embedded systems.**

**In embedded systems,** reconfigurable computing (RC) and reconfigurable platforms have become mainstream years ago[2] for accelerator use, flexibility, low cost, and, low power dissipation. Hundreds of Conferences focus on RC (fig. 3). Since about 2 years RC also goes rapidly into Supercomputing [8], and other HPC (High Performance Computing) to obtain massive speed-up by the fundamental paradigm shift coming along with RC: and, the personal supercomputer (PS) is near [9]. Networks of PS (NOPS) open new horizons of yet unbelievable dimensions in supercomputing performance [9].

**Reconfigurable Computing (and computational biology) make CS more fascinating — not only for students.**

**Reconfigurable Computing ignored** by our curricula is the wrong road map. RC is found practically everywhere which is illustrated by the reply of Google to the main keywords. „FPGA“ yields almost 5 million hits, and „Reconfigurable Computing“ more than 250,000 hits (fig. 1). RC goes into every application area (fig. 2). However, typical CS graduates are not familiar with using such platforms.

**HPC and Supercomputing are going reconfigurable.**

| FPGA and ... | Google |
|---|---|
| ... embedded | 3,280,000 |
| ... wireless | 1,490,000 |
| ... automotive | 915,000 |
| low power | 541,000 |
| ... medical | 710,000 |
| ... music | 398,000 |
| ... physics | 508,000 |
| ... chemical | 247,000 |
| ... defense | 287,000 |
| ... bio | 140,400 |
| ... weather | 128,000 |
| ... chemistry | 115,800 |
| ... molecular | 113,000 |
| high performance | 706,000 |
| supercomputing | 65,800 |
| n body problem | 28,200 |
| ... oil and gas | 22,300 |

Fig. 2. (Jan. 2006) Going to every application area.

**Many CS departments are obsolete.** In a speech at a summit meeting of US governors Bill Gates said: "American high schools are obsolete. Our high schools - even working exactly as designed - cannot teach our kids what they need to know today. The high schools of today teach kids about today's computers like on a 50-year-old mainframe. Our high schools were designed 50 years ago to meet the needs of another age. Without re-design for the needs of the 21st century, we will keep limiting - even ruining - the lives of millions of Americans every year." These statements also hold for many universities.

---

1.) FPGA stands for „Field-Programmable Gate Array“

2.)Pervasiveness of Reconfigurable Computing, but also request my detailed jornal proposal from: reiner@hartenstein.de

**The Role of Accelerators.** Hardwired accelerators, the result of software-to-hardware migration, are found everywhere for speed-up by avoiding the problems given by the sequential nature of instruction-stream-based traditional computing. For instance, a PC cannot maintain its own display without support by an accelerator (graphics chip or board). Because of skyrocketing mask cost, design cost, and design time, ***software-to-configware***[1] ***migration*** for Reconfigurable Computing (RC) is an extremely important alternative method, where similar speed-up factors can be obtained as known from hardwired accelerators. Compared to classical instruction-stream-based computing, such ***RC is based on a different common model*** and a fundamentally different mind set, which is often stalled by massive educational deficits: ***the software / configware chasm***, even more drastic than the old hardware / software chasm, e. g. affecting software-to-hardware migration.

| key word | # of hits |
|---|---|
| FPGA & conference | 1,070,000 |
| FPGA & workshop | 536,000 |
| FPGA & symposium | 412,600 |
| conference & "Reconfigurable Computing" | 126,000 |
| workshop & "Reconfigurable Computing" | 79,500 |
| symposium & "Reconfigurable Computing“ | 74,600 |

Fig. 3. Conferences by Google (Jan. 2006)

**Reconfigurable Computing now went into every application area.**

**(Structurally) Programmable Accelerators.** RC means the replacement of hardwired accelerators by (structurally) programmable platforms, which migrates the definition of wiring patterns and operator specs from before fabrication to the customer’s location after delivery.

**The new common model.** The von-Neumann paradigm (vN) is obsolete: the model of the mainframe era. Embedded systems are dominated by the symbiosis of the vN paradigm and the anti machine, which <u>is not</u> instruction-stream-driven. Both, hardwired and programmable (reconfigurable) accelerators can be modeled by the anti machine paradigm.

| Google vs. curriculum recommendations[a] | | | |
|---|---|---|---|
| key word | Google | ACM/IEEE 2004 [10] | ARTIST [12] consortium |
| FPGA | 9,960,000 | 0 | 3 |
| reconfigurable | 4,140,000 | 0 | 1 |
| Reconfigurable Computing | 256,000 | 0 | 0 |
| reconfigurable logic | 126,000 | 0 | 0 |
| configware | 13,400 | 0 | 0 |

a). search in all recommendation documents by „find and replace“ tool

Fig. 4. Illustrating gaps in curriculum recommendations.

**Going into Every Application Area** (fig. 2). Many years ago the use of reconfigurable platforms went from niche technology to mainstream. DaimlerChrysler, for instance, has a contract with Xilinx for creating FPGA architectures for automotive applications. Los Alamos National Laboratory has developed a FPGA-based self-repairing computing system scheduled for being launched into orbit. Recently Cray Inc. has introduced a supercomputing module including a FPGA-based accelerator. MAPLD, a special conference serves the needs of NASA and military applications for reconfigurable platforms lists much more very active application areas of reconfigurable platforms. Also the call for papers of a very large number of other conferences list a wide variety of application areas.

**The obsolescence of the von-Neumann-only common model drastically stalls progress.**

**The emerging Configware Industry.** Using software is RAM-based, which is the secret of success of the software industry. The RAM provides the flexibility. We have ***a second RAM-based source: configware.*** Supporting reconfigurable computing and reconfigurable logic, an emerging configware industry is already growing. Not being instruction-stream-based, configware is fundamentally different from software. ***Configware engineering is the counterpart of software engineering.*** Also configware code can be compiled e. g. from C language sources [9].

**Our curriculum recommendations fully ignore embedded systems’s strategically most important platform.**

**Ignored by our curriculum recommendations.** All this is still ignored even by newer well respected CS-related curriculum recommendations [10], where the number of encounters of all extremely important RC-related keywords is zero (fig. 4). For more recommendations also see „Artist FP5“ [11] [12].

**Configware Engineering is the fascinating counterpart of Software Engineering.**

**An update of curriculum recommendations is overdue.** There is an urgent need to elaborate a new roadmap for CS and related curricula. We need a dual-paradigm teaching methodology[2], going thoughout all stages of programs: from freshmen to graduates. A side effect of this bridging the software / configware gap would also help to bridge the old software / hardware gap. A rich supply of literature

1.) Configware, not instruction-stream-based, is the programming source for Reconfigurable Computing platforms.
2.)A „co-education“ based on the a symbiosis of vN and anti machine paradigm

is available for upgrading existing courses - more to change the point of view, rather than for swapping major parts of the contents. See appendix

## Literature


[1] P. Denning, A. McGettrick: Recentering Computer Science; Comm. ACM 48, 11 (Nov. 2005), 15-19
[2] D. Patterson: Restoring the Popularity of Computer Science; Comm. ACM 48, 9 (Sept. 2005), 25-28
[3] N.N.: U.S. losing lead in science and engineering (a study); Reuters, Washington, DC, July 8, 2005
[4] D. Kapur, J. McHale: Are We Losing the Global Race for Talent? Wall Street Journal (11/21/05), p. A17
[5] J. Vegso: Interest in CS as a major drops among incoming freshmen; Computing Research News 17, 3 (May 2005); http://www.cra.org/CRN/articles/may05/vegso
[6] P. Denning, The field of programmers myth; ACM 47, 7 (July 2004), 15-20
[7] F. Rammig: Visions from the IT Engine Room; http://www.ifip.org/secretariat/tc_visions/tc10_visions.htm
[8] N.N. (News Brief): Researchers Build Reconfigurable Supercomputer; COMPUTER, August 2005
[9] R. Hartenstein: Configware für Supercomputing: Aufbruch zum Personal Supercomputer; PIK 28, 4, (Dec. 2006)
[10] N.N.: Computing Curricula 2004; Joint Task Force for Computing Curricula 2004, 22 November 2004, etc.
[11] P. Caspi, A. Sangiovanni-Vincentelli, et al.: Guidelines for a Graduate Curriculum on Embedded Software and Systems; Workshop on Embedded Systems Education (WESE 2005), Sept. 22, 2005, Jersey City, New Jersey, USA
[12] N.N.: W2.All.Y1Guidelines for a Graduate Curriculum on Embedded Software and Systems; ARTIST Consortium, May 12, 2003, http://www.artist-embedded.org/Education/Education.pdf


# Appendix

*This appendix drafts also a kind of road map to urgently needed improvements of our curricula, also CS, CE, and ECE curricula - for graduate and undergraduate programs as well as continuing education.*

### Motivations of the 1st International Workshop on Reconfigurable Computing Education[1]

Being the fastest growing segment of the microelectronics market, FPGAs have become mainstream already years ago in all kinds of embedded systems. More recently FPGAs and other Reconfigurable Computing (RC) platforms are rapidly moving into practically every application area, such as automotive, aerospace, defense, medical, chemistry, molecular biology, physics, astrophysics, high performance computing, supercomputing, and many other areas.

**Fragmentation.** Each of these application domains has only a limited view of computing and takes it more as a mere technique than as a science on its own. Consequences are, that it makes it very difficult to bridge the cultural and practical gaps. Given this fragmentation, it can be rather hard to investigate, since there are so many different actors and departments involved. Including and programming reconfigurable platforms in the design of embedded systems as well as embedded real-time systems and all other application areas requires more skills at least from computer sciences. Currently it requires to involve experts from different backgrounds, with dissenting points of view, not only for test and verification of such designs, if at all possible, being very expensive and delaying significantly the introduction of products.

**Productivity crisis.** Rapidly growing complexity and pervasiveness of RC-based multi-paradigm devices leads to a productivity crisis of major proportions. On the other hand RC is an efficient approach to cope with the accelerating VLSI design crisis. While the economic importance of RC and its FPGAs is widely acknowledged, the strategic dimension of RC has not been appreciated until recently, academia has failed to pay sufficient attention to the education of a community of high-quality system designers and configware programmers using such platforms. This has motivated a recent but ever growing interest in the question of educating specialists in this domain and this has also been recognized as a particularly difficult problem.

**Need for unifying the foundations.** We need to counter the current trend, where specialization is the target of education systems. We need to go toward interdisciplinary CS-related curricula for unifying the foundations of the discipline since it has become evident that fundamental problems are shared across several different application domains. It is the goal of this first workshop to bring together researchers, educators, and industrial representatives to share design, research, and education experiences in Reconfigurable Computing and a wide variety of its applications. RC-based design insolves not only hardware-software co-design. What is really needed is a much more interdisciplinary approach of hardware-configware-software co-design, not only as a design practice, but also as part of CS, CE, and EE curricula.

**New educational approaches.** Although configware engineering is a discipline of its own, fundamentally different from software engineering, and, a configware industry is already existing and growing, it is too often ignored by our curricula. Modern FPGAs as COTS (commodities off the shelf) have all 3 paradigms on board of the same VLSI chip: hardwired accelerators, microprocessors (and memory banks), and FPGAs, and we need software and configware to program the same chip. To cope with the clash of cultures we need interdisciplinary curricula merging all these different backgrounds in a systematic way. We need innovative lectures and lab courses supporting the integration of reconfigurable computing into progressive curricula. The workshop intends to provide a forum for presenting experiences and new educational approaches and for discussing the pros and cons.

1.) http://helios.informatik.uni-kl.de/RCeducation/